

Exigences Non Fonctionnelles

DAFA NRAF

3.5.0

1	Exigences	11
1.1	Exigences non fonctionnelles	11
1.1.1	A ventiler	11
1.1.2	Contraintes Documentaires	11
1.1.2.1	Documentation technique	11
1.1.2.1.1	Documentation technique - Mise en forme et exportabilité	11
1.1.2.1.2	Documentation technique - Approche "doc as code"	12
1.1.2.1.3	Documentation technique - Format	12
1.1.2.1.4	Documentation technique - Stockage de la documentation relative composant	12
1.1.2.1.5	Documentation technique - Stockage de la documentation transverse	12
1.1.2.1.6	Documentation technique - Modélisation	13
1.1.2.2	Manuel de Dev & Intégration	13
1.1.2.2.1	Manuel DEV/INT : Modèle Manuel DEV/INT	13
1.1.2.2.2	Manuel DEV/INT - Generation Fichier Word	13
1.1.2.2.3	Manuel DEV/INT : 00 - Header.md	14
1.1.2.2.4	Manuel DEV/INT - 01 - Preamble.md - Objet du document	14
1.1.2.2.5	Manuel DEV/INT - 01 - Preamble.md - Documents Applicables	14
1.1.2.2.6	Développement	14
1.1.2.2.6.1	Manuel DEV/INT - 02 - Developpement.md - Starter Kit FullStack	15
1.1.2.2.6.2	Manuel DEV/INT - 02 - Developpement.md - Outils de développement	15
1.1.2.2.6.3	Manuel DEV/INT - 02 - Developpement.md - Stockage et récupération des sources	15
1.1.2.2.6.4	Manuel DEV/INT - 02 - Developpement.md - Workflow de développement	15
1.1.2.2.6.5	Manuel DEV/INT - 02 - Developpement.md - Configuration applicative	16
1.1.2.2.6.6	Manuel DEV/INT - 02 - Developpement.md - Bouchonnage des applications	16
1.1.2.2.6.7	Manuel DEV/INT - 02 - Developpement.md - Compilation	16
1.1.2.2.6.8	Manuel DEV/INT - 02 - Developpement.md - Exécution sur le poste de	17
1.1.2.2.6.9	Manuel DEV/INT - 02 - Developpement.md - Exécution sur le poste de	17
1.1.2.2.6.10	Manuel DEV/INT - 02 - Developpement.md - Tests Unitaires	17
1.1.2.2.6.11	Manuel DEV/INT - 02 - Developpement.md - Tests des API avec Postman	18
1.1.2.2.6.12	Manuel DEV/INT - 02 - Developpement.md - Tests des messages KAFKA avec	18
1.1.2.2.7	Intégration	18
1.1.2.2.7.1	Manuel DEV/INT - 03 - Integration.md - Rappels du modèle de branche UCN	18
1.1.2.2.7.2	Manuel DEV/INT - 03 - Integration.md - Chaines de construction	19
1.1.2.2.7.3	Manuel DEV/INT - 03 - Integration.md - Mesures de la qualité	19
1.1.2.2.7.4	Manuel DEV/INT - 03 - Integration.md - Deploiement	20
1.1.2.2.7.5	Manuel DEV/INT - 03 - Integration.md - Deploiement Plateforme HAWAI	20
1.1.2.2.7.6	Manuel DEV/INT - 03 - Integration.md - Deploiement Plateforme SX5	20
1.1.2.2.7.7	Manuel DEV/INT - 03 - Integration.md - Deploiement Plateforme PFS	21
1.1.2.2.7.8	Manuel DEV/INT - 03 - Integration.md - Deploiement des Bundles Archimed	21
1.1.2.2.7.9	Manuel DEV/INT - 03 - Integration.md - Tests d'intégration - accès aux services	21
1.1.2.2.7.10	Manuel DEV/INT - 03 - Integration.md - Tests d'intégration - exécution des tests	22
1.1.2.2.7.11	Manuel DevInt/SNV2 - Chapitre traitant de l'exploitation sur SNV2	22

1.1.2.2.7.12	Manuel DevInt/HAWAI - Chapitre traitant de l'exploitation sur Hawai	22
1.1.2.2.7.13	Manuel DevInt/PFS - Chapitre traitant de l'exploitation sur PFS	22
1.1.2.3	Spécifications Techniques Détaillées	23
1.1.2.3.1	STD - Modèle STD DACStack	23
1.1.2.3.2	STD - Génération document word	23
1.1.2.3.3	STD - 00 - Header	23
1.1.2.3.4	STD - 01 - Preamble - Objet du document - Objet du document.....	24
1.1.2.3.5	STD - 01 - Preamble - Documents Applicables.....	24
1.1.2.3.6	STD - 02 - Module - Récapitulatif - Contexte	24
1.1.2.3.7	STD - 02 - Module - Récapitulatif - Périmètre de la solution.....	24
1.1.2.3.8	STD - 03 - Rappel de spécifications générales	25
1.1.2.3.9	STD - 03 - Rappel de spécifications générales - Organisation des sources - Librairies - Internes	25
1.1.2.3.10	STD - 03 - Rappel de spécifications générales - Organisation des sources - Librairies - Externes	25
1.1.2.3.11	STD - 03 - Rappel de spécifications générales - Organisation des sources - Logiciels sur étagère	25
1.1.2.3.12	STD - 03 - Rappel de spécifications générales - Organisation des sources - Logiciels sur étagère (licences)	26
1.1.2.3.13	STD - 05 - Spécifications du module - Vue d'ensemble - Vue des composants et des services du module.....	26
1.1.2.3.14	STD - 05 - Spécifications du module - Vue d'ensemble - Vue des fonctionnalités portées par le module	27
1.1.2.3.15	STD - 05 - Spécifications du module - Vue d'ensemble - Vue des ressources portées par le module	27
1.1.2.3.16	STD - 05 - Spécifications du module - Détail des composants.....	27
1.1.2.3.17	STD - 05 - Spécifications du module - Détail des composants - Composant Bundle Archimed	27
1.1.2.3.18	STD - 05 - Spécifications du module - Détail des composants - Composant Base de Données	28
1.1.2.3.19	STD - 05 - Spécifications du module - Détail des composants - Composant Topic Kafka	28
1.1.2.3.20	STD - 05 - Spécifications du module - Détail des composants - Composant API - Diagramme de classe	29
1.1.2.3.21	STD - 05 - Spécifications du module - Détail des composants - Composant API - Enumérations	29
1.1.2.3.22	STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Mapping Table-Entité	29
1.1.2.3.23	STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Mapping Entité-DTO.....	30
1.1.2.3.24	STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Mapping Message KAFKA-DTO.....	30
1.1.2.3.25	STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Génération fichier	30
1.1.2.3.26	STD - 05 - Spécifications du module - Détail des composants - Composant API - Services	30
1.1.2.3.27	STD - 06 - Interfaces - Schéma des échanges	31
1.1.2.3.28	STD - 06 - Interfaces - Interfaces fournies	31

1.1.2.3.29	STD - 06 - Interfaces - Interfaces consommées	31
1.1.2.3.30	STD - 06 - Interfaces - Bus de messages KAFKA	32
1.1.2.3.31	STD - 06 - Interfaces - Traitement de Fichier - Bucket S3	32
1.1.2.3.32	STD - 06 - Interfaces - Supervision.....	32
1.1.2.3.33	STD - 07 - Intégration.....	33
1.1.2.4	Dossier d'Exploitation.....	33
1.1.2.4.1	DEX - Modèle DEX.....	33
1.1.2.4.2	DEX - Procédures d'exploitations	34
1.1.2.4.3	DEX - Métriques et supervision	34
1.1.2.4.4	DEX - Sondes.....	34
1.1.2.4.5	DEX - Composants Applicatifs.....	34
1.1.2.4.6	DEX - Outillage.....	35
1.1.2.4.7	DEX - Tâches planifiées	35
1.1.2.5	Documentation générée par GLD.....	35
1.1.2.5.1	Livraison/SNV2 - Versioning documentation dans GLD	35
1.1.2.5.2	Livraison/SNV2 - Déclaration de la modification d'une base de données dans GLD	36
1.1.2.5.3	Backend/SNV2 - Modification de traitements/services	36
1.1.2.5.4	Livable - Exigences fonctionnelles du GLD.....	36
1.1.2.5.5	Livable/SNV2 - Fiche de planification	36
1.1.2.5.6	Livable/SNV2 - Documentation GLD	37
1.1.2.6	Doc de lot / Install	37
1.1.2.6.1	Doc de lot - Format.....	37
1.1.2.6.2	Doc de lot - Granularité.....	37
1.1.2.7	Changelog	37
1.1.2.7.1	Changelog	38
1.1.2.8	DocAsCode.....	38
1.1.3	Contraintes de Conception.....	38
1.1.3.1	Conception - Identification des ressources par UUID	39
1.1.3.2	Conception - Suppression logique.....	39
1.1.3.3	Conception - API.....	40
1.1.3.4	Conception - Mutualisation des sources Agent et API	40
1.1.3.5	Conception - Scallabilité.....	40
1.1.3.6	Conception - Mutualisation.....	40
1.1.3.7	Conception - NullPointer	41
1.1.3.8	Conception - Verrous optimistes	41
1.1.4	Contraintes de Développement.....	41
1.1.4.1	Documents applicables	41
1.1.4.2	Composants applicatifs	42
1.1.4.2.1	Backend	42
1.1.4.2.1.1	Généralités.....	42
1.1.4.2.1.1.1	COTS - Version du Jdk	42
1.1.4.2.1.1.2	Backend - Utilisation des Enum.....	42
1.1.4.2.1.1.3	Backend - Utilisation des Bucket S3.....	43
1.1.4.2.1.1.4	Backend - Stockage des fichiers	43
1.1.4.2.1.1.5	Backend - Pom parent.....	43
1.1.4.2.1.2	API	43
1.1.4.2.1.2.1	API Backend - API Rest	44
1.1.4.2.1.2.2	API Backend - Compatibilité des API avec OData.....	44

1.1.4.2.1.2.3	API Backend - Certificats et exposition Archimed	44
1.1.4.2.1.2.4	API Backend - Consommation de services externes	45
1.1.4.2.1.2.5	API Backend - Localisation des contrats de service exposés	45
1.1.4.2.1.2.6	API Backend - Récupération des contrats de services consommés	45
1.1.4.2.1.2.7	API Backend - Granularité des API	45
1.1.4.2.1.2.8	API Backend - Pagination des listes.....	46
1.1.4.2.1.2.9	API Backend - OData requêtes complexes en deux appels	46
1.1.4.2.1.2.10	API Backend - Consommation de services externes - cache et throttling.....	47
1.1.4.2.1.2.11	API Backend - Ressources historisées	47
1.1.4.2.1.3	Batch.....	49
1.1.4.2.1.3.1	Batch - Framework.....	49
1.1.4.2.1.3.2	Batch - Planification batch multi applicatifs.....	49
1.1.4.2.1.3.3	Batch/SNV2 - Planification batch régionaux	50
1.1.4.2.1.3.4	Batch/SNV2 - Création commande	50
1.1.4.2.1.3.5	Batch/PFS - Planification batch nationaux via Automator	50
1.1.4.2.1.3.6	Batch/Hawai - Planification.....	50
1.1.4.2.1.3.7	Batch backend/PFS - Déclaration dans le charthelm.....	51
1.1.4.2.1.3.8	Batch backend/PFS - Configuration environnement dev/autre	51
1.1.4.2.2	Frontend.....	51
1.1.4.2.2.1	Frontend - Mise en place de BIPE	51
1.1.4.2.3	Base de données.....	52
1.1.4.2.3.1	Base de données/SNV2 - Evolution de schéma	52
1.1.4.2.3.2	Base de données/SNV2 - Nature de la base de données - Migration	52
1.1.4.2.3.3	Base de données/SNV2 - Utilisation de fonctions standards - Migration.....	52
1.1.4.2.3.4	Base de données/SNV2 - Bascule des SGDB - Migration	53
1.1.4.2.3.5	Base de donnée/PFS - Evolution de schéma	53
1.1.4.2.3.6	Base de données - Procédure de restauration	53
1.1.4.2.3.7	Base de données - Configuration de la disponibilité	53
1.1.4.2.3.8	Base de données - Procédure de backup.....	53
1.1.4.2.3.9	Base de donnée/PFS - Ouverture du port.....	54
1.1.4.2.3.10	Base de donnée/PFS - PostgREST.....	54
1.1.4.2.3.11	Base de données - Procédure de restauration_Copie_1	54
1.1.4.2.4	Bus de message.....	54
1.1.4.2.4.1	Bus de message - Utilisation Kafka	55
1.1.4.2.4.2	Bus de message - Se connecter à Kafka de manière sécurisée	55
1.1.4.2.4.3	Bus de message - Définition des topics	55
1.1.4.2.4.4	Bus de message - Variabilisation des topics.....	55
1.1.4.2.4.5	Bus de message - Sécuriser l'accès aux topics en utilisant les ACL.....	56
1.1.4.2.4.6	Bus de messages - Format des messages.....	56
1.1.4.2.4.7	Bus de messages - Structure d'un messages fonctionnel.....	56
1.1.4.2.4.8	Bus de messages - Désérialisation des messages.....	57
1.1.4.2.4.9	Bus de messages - Versioning des messages	57
1.1.4.2.4.10	Bus de messages - Stratégie de développement	58
1.1.4.2.4.11	Bus de message/Inter-domaine - Ajout fournisseur.....	58
1.1.4.2.4.12	Bus de message/Inter-domaine - Abonnement à un topic sensible	58
1.1.4.2.4.13	Bus de messages/Inter-domaine - Vérification de grammaire.....	58
1.1.4.2.4.14	Bus de message/Inter-domaine - Accès topic sécurisé	59
1.1.4.2.4.15	Bus de message/Intra-domaine - Accès topic	59

1.1.4.2.4.16	Bus de message/Intra-Domaine - Accès topic sensible	59
1.1.4.2.4.17	Bus de messages - Volumétrie.....	59
1.1.4.2.4.18	Bus de messages - Publication de type Ressource	60
2	Règles de déclenchement	60
2.1.1	Message de publication « Ressource ».....	60
2.1.1.1.1.1	Bus de messages - Header technique	60
2.1.1.1.2	Transverse	61
2.1.1.1.2.1	Observabilité	61
2.1.1.1.2.1.1	Log - Framework de gestion des logs.....	61
2.1.1.1.2.1.2	Log/PFS - Gestion des logs	61
2.1.1.1.2.1.3	Log/SNV2 - Gestion des logs	61
2.1.1.1.2.1.4	Log - Niveaux de sévérité.....	62
2.1.1.1.2.1.5	Log - Langue des messages	62
2.1.1.1.2.1.6	Log - Message techniques sans informations personnelles.....	62
2.1.1.1.2.1.7	Log - Contextualisation du message	63
2.1.1.1.2.1.8	Log - Ségrégation des messages	63
2.1.1.1.2.1.9	Configuration - Niveau de sévérité des log	63
2.1.1.1.2.1.10	Configuration - Chargement à chaud du niveau de log	63
2.1.1.1.2.1.11	Supervision - Publication des métriques supervisées.....	64
2.1.1.1.2.1.12	Supervision - Métriques suivies par la supervision	64
2.1.1.1.2.1.13	Supervision/PFS - Outillage de supervision	64
2.1.1.1.2.1.14	Log - Log minimum	64
2.1.1.1.2.1.15	Supervision/PFS - Healthcheck.....	65
2.1.1.1.2.2	Qualité.....	66
2.1.1.1.2.2.1	Qualité - Outillages.....	66
2.1.1.1.2.2.2	Qualité - Langage français	66
2.1.1.1.2.2.3	Qualité - Pas de code mort.....	67
2.1.1.1.2.2.4	Qualité - Documentation du code	67
2.1.1.1.2.2.5	Qualité - Informations minimales contenues dans la description des fonctions.....	67
2.1.1.1.2.2.6	Qualité/Hawai - Encodage sources	67
2.1.1.1.2.2.7	Qualité - Traçage des exclusions	67
2.1.1.1.2.2.8	Qualité - Checkstyle	68
2.1.1.1.2.2.9	Qualité - Gestion des Exception	68
2.1.1.1.2.3	Testabilité.....	68
2.1.1.1.2.3.1	Testabilité - Création de bouchon.....	68
2.1.1.1.2.3.2	Testabilité - Packaging des bouchons	69
2.1.1.1.2.3.3	Testabilité - Intégration Backend postman.....	69
2.1.1.1.2.3.4	Testabilité - Nommage des Tests Unitaires	69
2.1.1.1.2.4	Configurabilité	70
2.1.1.1.2.4.1	Configuration - Gestion par environnements d'exécution	70
2.1.1.1.2.4.2	Configuration - Variabilisation dépendant de l'environnement d'exécution	70
2.1.1.1.2.4.3	Configuration - Méta informations dans entêtes pour routing archimed	70
2.1.1.1.2.4.4	Configuration - Activation Prisme	71
2.1.1.1.2.4.5	Configuration - Propriétés dans les entêtes HTTP.....	71
2.1.1.1.2.5	COTS.....	71
2.1.1.1.2.5.1	COTS - Licence Open Source non contaminante.....	71
2.1.1.1.2.6	Sécurité	72
2.1.1.1.2.6.1	Sécurité - Analyse des vulnérabilités durant tout le cycle de vie du produit	72

2.1.1.1.2.6.2	Sécurité - Appels HTTPS	72
2.1.1.1.2.6.3	Sécurité - Charte	72
2.1.1.1.2.7	Transverse - Utilisation des templates FullStack	73
2.1.1.1.2.8	Transverse - Environnement d'exécution sur le poste de développement	73
2.1.1.1.3	Kubernetes	73
2.1.1.1.3.1	Chart Helm - Sondes	74
2.1.1.2	Gestion de configuration	74
2.1.1.2.1	Dépôts	74
2.1.1.2.1.1	Dépôt - Nomenclature des noms de dépôt	74
2.1.1.2.1.2	Dépôt - Groupe Gitlab	75
2.1.1.2.1.3	Dépôt - Création du dépôt pour les prototypes	75
2.1.1.2.1.4	Dépôt/Bouchon - Création des dépôts pour les bouchons	75
2.1.1.2.1.5	Dépôt/Backend - Création des dépôts pour composant Backend	75
2.1.1.2.1.6	Dépôt/FrontEnd - Création des dépôts pour composant Frontend	75
2.1.1.2.1.7	Dépôt/Bundle Archimed - Création des dépôts de déploiement d'un bundle Archimed Backend	76
2.1.1.2.1.8	Dépôt/Bundle Archimed - Création des dépôts de déploiement d'un bundle Archimed Frontend	76
2.1.1.2.1.9	Dépôt/Chart Helm - Création des dépôts de création du Chart Helm sur PFS	76
2.1.1.2.1.10	Dépôt/Chart Helm - Création des dépôts de déploiement du Chart Helm	77
2.1.1.2.1.11	Dépôt - Création d'un dépôt documentaire	77
2.1.1.2.2	Modèle de branchement	77
2.1.1.2.2.1	Modèle de branchement - Traçabilité des tâches	78
2.1.1.2.2.2	Modèle de branchement - Messages de commit	78
2.1.1.2.2.3	Modèle de branchement - Création des branches de travail	78
2.1.1.2.2.4	Modèle de branchement - Nommage des branches de travail	78
2.1.1.2.2.5	Modèle de branchement - Rapatriement d'un travail sur develop	79
2.1.1.2.2.6	Modèle de branchement - GitFlow	79
2.1.1.3	Comptes techniques	80
2.1.1.3.1	Compte technique - Déclenchement procédure CADRAN	80
2.1.1.3.2	Compte technique - Accès au portail technique	80
2.1.1.3.3	Compte technique/PFS - Accès à la PFS de developement	80
2.1.2	Contraintes de Construction	81
2.1.2.1	Documents applicables	81
2.1.2.2	Artefacts construits	81
2.1.2.2.1	Artefact Maven	81
2.1.2.2.1.1	Artefact Maven - Nommage artifactId	81
2.1.2.2.1.2	Artefact Maven - Contract de service exposé	82
2.1.2.2.1.3	Artefact Maven - Caractéristiques	82
2.1.2.2.1.4	Configuration - Construction artefact maven	82
2.1.2.2.2	Bundle Archimed	83
2.1.2.2.2.1	Bundle Archimed - Cohérence contract/règles de sécurité	83
2.1.2.2.2.2	Bundle Archimed - Validation des règles de sécurité	83
2.1.2.2.2.3	Bundle Archimed - Localisation des contrats des services	83
2.1.2.2.2.4	Bundle Archimed - Nommage d'un bundle backend	84
2.1.2.2.2.5	Bundle Archimed - Structure d'un bundle backend	84
2.1.2.2.2.6	Bundle Archimed - Nommage d'un bundle frontend	84
2.1.2.2.2.7	Bundle Archimed - Structure d'un bundle frontend	84

2.1.2.2.2.8	Bundle Archimed - Métainformations à ajouter aux swaggers	85
2.1.2.2.2.9	Bundle Archimed/SNV2 - Métainformations à ajouter aux swaggers	85
2.1.2.2.2.10	Bundle Archimed/PFS - Métainformations à ajouter aux swaggers	85
2.1.2.2.2.11	Bundle Archimed - Version.....	86
2.1.2.2.2.12	Bundle Archimed - Règles de sécurité	86
2.1.2.2.3	Chart Helm (PFS)	86
2.1.2.2.3.1	Chart Helm - Structure.....	86
2.1.2.2.3.2	Chart Helm - Nommage du chart	87
2.1.2.2.3.3	Chart Helm - Version.....	87
2.1.2.2.3.4	Chart Helm - Variabilisation des templates	87
2.1.2.2.3.5	Chart Helm - Variabilisation des fichiers de configuration des composants	88
2.1.2.2.3.6	Chart Helm - Base de données	88
2.1.2.2.3.7	Chart Helm - Volumétrie	88
2.1.2.2.3.8	Chart Helm - Exposition des services	88
2.1.2.2.3.9	Chart Helm - Configurabilité de la sécurité	89
2.1.2.2.3.10	Chart Helm - Configurabilité des ressources	89
2.1.2.2.3.11	Chart Helm - Configurabilité d'accès aux services externes	89
2.1.2.2.3.12	Chart Helm - Gestion mots de passe.....	90
2.1.2.2.3.13	Chart Helm - Déploiement.....	90
2.1.2.2.3.14	Chart Helm - Dépôt de stockage	90
2.1.2.2.3.15	Chart Helm - Exposition de la base de donnée	91
2.1.2.2.3.16	Chart Helm - Polices de sécurité	91
2.1.2.2.3.17	Chart Helm - Operateurs et subcharts.....	91
2.1.2.2.4	Image Docker	91
2.1.2.2.4.1	Image Docker - Nommage	92
2.1.2.2.4.2	Image Docker - Versioning	92
2.1.2.2.4.3	Image Docker - Configurabilité	92
2.1.2.2.4.4	Image Docker - Environnement de développement	92
2.1.2.2.4.5	Image Docker - Dépôt de stockage	93
2.1.2.2.4.6	Configuration - Génération des Images Docker	93
2.1.2.2.4.7	Image Docker - Configuration de la timezone	93
2.1.2.3	Chaines de construction.....	93
2.1.2.3.1	Généralités	94
2.1.2.3.1.1	Chaîne de construction - Utilisation de Jenkins.....	94
2.1.2.3.1.2	Chaîne de construction - Fourniture de l'intégration par composant	94
2.1.2.3.1.3	Chaîne de construction - Construction des artefacts ACROSS	94
2.1.2.3.1.4	Chaîne de construction - Caractéristiques	94
2.1.2.3.1.5	Chaîne de construction - Notification en cas d'erreur.....	95
2.1.2.3.1.6	Chaîne de construction - Variables d'environnement	95
2.1.2.3.1.7	Chaîne de construction/Composant - Création/Déploiement composant applicatif ..	95
2.1.2.3.1.8	Chaîne de construction/Bundle Archimed - Création/Installation Bundle Backend ..	96
2.1.2.3.1.9	Chaîne de construction/Bundle Archimed - Création/Installation Bundle FrontEnd ..	96
2.1.2.3.1.10	Chaîne de construction/Chart Helm - Création	96
2.1.2.3.1.11	Chaîne de construction/Chart Helm - Déploiement sur environnement exécution..	96
2.1.2.3.1.12	Configuration - Définition des chaînes de construction	97
2.1.2.3.2	Sécurité	97
2.1.2.3.2.1	Chaîne de construction/Sécurité - Outil de contrôle des dépendances	97

2.1.2.3.2.2	Chaîne de construction/Sécurité - Seuil d'acceptabilité des violations de dépendances	97
2.1.2.3.2.3	Chaîne de construction/Sécurité - Outil d'analyse des vulnérabilités	98
2.1.2.3.2.4	Chaîne de construction/Sécurité - Seuil d'acceptabilité des violations des vulnérabilités	98
2.1.2.3.2.5	Chaîne de construction/Sécurité - Mise à disposition du rapport de vulnérabilité	98
2.1.2.3.3	Qualité	98
2.1.2.3.3.1	Chaîne de construction/Composant - Barrière qualité	99
2.1.2.3.3.2	Chaîne de construction/Composant - Violations de qualités	99
2.1.2.3.3.3	Chaîne de construction/Composant - Mise à disposition du rapport qualité	99
2.1.2.3.3.4	Qualité - Règles de qualité TypeScript	100
2.1.2.3.3.5	Qualité - Règles de qualités Java	100
2.1.2.3.3.6	Qualité - Seuils autorisés	100
2.1.2.3.3.7	Qualité - Vérification des scripts shell	100
2.1.3	Contraintes de Livraison	101
2.1.3.1	Documents applicables	101
2.1.3.2	Livrables	101
2.1.3.2.1	Livable - Rapport des tests usines	101
2.1.3.2.2	Livable - Procédure de migration	102
2.1.3.2.3	Livable/SNV2 - Paramétrage des applications SNV2 (variables Castor)	102
2.1.3.2.4	Livable/SNV2 - Politiques Axway	102
2.1.3.2.5	Livable/HAWAI - Nature du livrable	102
2.1.3.2.6	Livable/HAWAI - Description des target	103
2.1.3.2.7	Livable/HAWAI - Description base de données	103
2.1.3.2.8	Livable/HAWAI - Description des propriétés applicables	103
2.1.3.2.9	Livable/HAWAI - Cohérence des versions	103
2.1.3.2.10	Livable/HAWAI - Localisation du stockage du DIT/DDP	104
2.1.3.2.11	Livable/HAWAI - Localisation DEX	104
2.1.3.2.12	Livable/HAWAI - Localisation spécification fonctionnelle	104
2.1.3.2.13	Livable/HAWAI - Politiques Axway	104
2.1.3.2.14	Livable/HAWAI - Document décrivant les composants	105
2.1.3.3	Livraison	105
2.1.3.3.1	Livraison - Traçabilité d'une livraison (Redmine)	105
2.1.3.3.2	Livraison - Déversement des sources dans le dépôt URSSAF	106
2.1.3.3.3	Livraison - Lancement des chaînes de construction	106
2.1.3.3.4	Livraison/HAWAI - Traçabilité d'une livraison (Redmine)	107
2.1.3.3.5	Livraison/HAWAI - Mise à disposition du livrable (Redmine)	107
2.1.3.3.6	Livraison/HAWAI - Nomenclature du nom du lot	107
2.1.3.3.7	Livraison/SNV2 - Déversement des Scripts migration	108
2.1.3.3.8	Livraison/SNV2 - Traçabilité de la livraison de Batchs (Redmine)	108
2.1.3.3.9	Livraison/SNV2 - Traçabilité de la livraison des containers (Redmine)	109
2.1.3.3.10	Livraison/SNV2 - Traçabilité de la livraison d'une Base de données (Redmine)	109
2.1.3.3.11	Livraison/PFS - Nomenclature du nom du lot	110
2.1.3.3.12	Livraison - Gestion des versions et tag git	110
2.1.3.3.13	Livraison/PFS - Gestion des versions	110
2.1.3.3.14	Livraison - Tracabilité des livraisons documentaires DocAsCode	111
2.1.3.3.15	Livraison - Déversement des sources - contrôles pre commits	111
2.1.3.4	Déploiement	111

2.1.3.4.1	Point de contrôle/SNV2 - Contrôle CTRL-1-Path-Image-1	112
2.1.3.4.2	Point de contrôle/SNV2 - Contrôle CTRL-2-Path-Image-2	112
2.1.3.4.3	Point de contrôle/SNV2 - Contrôle CTRL-3-Nom-Image -1	112
2.1.3.4.4	Point de contrôle/SNV2 - Contrôle CTRL-4-Version-Image-1	113
2.1.3.4.5	Point de contrôle/SNV2 - Contrôle CTRL-5-Nom-Image -2	113
2.1.3.4.6	Point de contrôle/SNV2 - Contrôle CTRL-6-Nom-Image-3	113
2.1.3.4.7	Point de contrôle/SNV2 - Contrôle CTRL-7-Name-Type-1	114
2.1.3.4.8	Point de contrôle/SNV2 - Contrôle CTRL-8-Erb-Nom	114
2.1.3.4.9	Point de contrôle/SNV2 - Contrôle CTRL-9-Erb-Encoding	114
2.1.3.4.10	Point de contrôle/SNV2 - Contrôle CTRL-10-Nom-Image-4	115
2.1.3.4.11	Point de contrôle/SNV2 - Contrôle CTRL-11-NomService-1	115
2.1.3.4.12	Point de contrôle/SNV2 - Contrôle CTRL-12-TypeVM-1	115
2.1.3.4.13	Point de contrôle/SNV2 - Contrôle CTRL-13-ImageProd-1	115
2.1.3.4.14	Point de contrôle/SNV2 - Contrôle CTRL-14-OnlyOn-1	116
2.1.3.4.15	Point de contrôle/SNV2 - Contrôle CTRL-15-Bascule-1	116
2.1.3.4.16	Point de contrôle/SNV2 - Contrôle CTRL-16-PlagePort-1	116
2.1.3.4.17	Point de contrôle/SNV2 - Contrôle CTRL-17-ImageDev-1	116
2.1.3.4.18	Point de contrôle/PFS - Contrôle nommage	116
2.1.4	Contraintes d'observabilité	117
2.1.5	Contraintes de Performance	117
2.1.5.1	Performance - Consommateur de messages	117
2.1.5.2	Performance - Webservice	117
2.1.5.3	Performance - IHM	118
2.1.5.4	Performance - Dépassement de limites	118
2.1.6	ENF Dépréciées ou Annulées	118
2.1.6.1	Backend/PFS - Communication inter-namespace	119
2.1.6.2	Backend/PFS - Communication intra-namespace	119
2.1.6.3	Performance	119
2.1.6.3.1	A - IHM	119
2.1.6.3.2	B - Webservices/API	119
2.1.6.3.3	C - Batch	120
2.1.6.3.3.1	RAF-PRF-Traitement de reprise inférieur à 24h	120
2.1.6.3.3.2	RAF-PRF-Taux de traitement pour synchronisation des écarts négatifs = 150 000/h 120	
2.1.6.3.3.3	RAF-PRF-Controle de la taille des FS exfv2 et bddv2 avant et après le traitement de reprise des données	121

1 Exigences

1.1 Exigences non fonctionnelles

Étiquette du champ	Valeur du champ
ID de l'exigence	776
Nom	Exigences non fonctionnelles
Etat exigence	18
Description	

1.1.1 A ventiler

Étiquette du champ	Valeur du champ
ID de l'exigence	1412
Nom	A ventiler
Etat exigence	27
Description	

1.1.2 Contraintes Documentaires

Étiquette du champ	Valeur du champ
ID de l'exigence	777
Nom	Contraintes Documentaires
Etat exigence	17
Description	

1.1.2.1 Documentation technique

Étiquette du champ	Valeur du champ
ID de l'exigence	778
Nom	Documentation technique
Etat exigence	12
Description	
Ceci est un test d'injection d'une exigence	

1.1.2.1.1 Documentation technique - Mise en forme et exportabilité

Étiquette du champ	Valeur du champ
ID de l'exigence	779
Nom	Documentation technique - Mise en forme et exportabilité
Etat exigence	52
Description	
Les documents utilisant AsciiDoc doivent respecter les bases du langage pour la mise en forme afin d'assurer un rendu correct vers le format word à l'aide du modèle et de l'outillage fourni dans le dépôt :	

<http://gitlab.altair.recouv/sded/coeur-de-metier-recouvrement/controle-et-recouvrement-amiable-force-parametres/raf/raf-indus/raf-dacstack>

1.1.2.1.2 Documentation technique - Approche "doc as code"

Étiquette du champ	Valeur du champ
ID de l'exigence	780
Nom	Documentation technique - Approche "doc as code"
Etat exigence	44
Description	L'approche "Documentation as code" doit être privilégiée pour tout document technique produit par le prestataire. Justification: Transformation aisée dans d'autres format (pdf, word ...) et visualisable directement dans GIT

1.1.2.1.3 Documentation technique - Format

Étiquette du champ	Valeur du champ
ID de l'exigence	781
Nom	Documentation technique - Format
Etat exigence	45
Description	Le format privilégié pour la documentation technique produite par le prestataire doit être AsciiDoc. Les points suivant doivent en particulier être respectés : • Pas de numérotation manuelle des chapitre, utilisation des niveaux de titre • Utilisation des balises adaptées pour les citations et extraits de code

1.1.2.1.4 Documentation technique - Stockage de la documentation relative composant

Étiquette du champ	Valeur du champ
ID de l'exigence	782
Nom	Documentation technique - Stockage de la documentation relative composant
Etat exigence	42
Description	La documentation technique d'un composant doit être stockée dans le dépôt GIT du composant.

1.1.2.1.5 Documentation technique - Stockage de la documentation transverse

Étiquette du champ	Valeur du champ
ID de l'exigence	783
Nom	Documentation technique - Stockage de la documentation transverse
Etat exigence	7
Description	Les documentations techniques communes à plusieurs projets ou non techniques doivent être stockées dans un dépôt GIT dédié.

--

1.1.2.1.6 Documentation technique - Modélisation

Étiquette du champ	Valeur du champ
ID de l'exigence	784
Nom	Documentation technique - Modélisation
Etat exigence	5
Description	Les diagrammes inclus dans la documentation doivent être réalisés avec un des outils suivants : - PlantUML - Archi Les sources des diagrammes devront être inclus avec le doc as code.

1.1.2.2 Manuel de Dev & Intégration

Étiquette du champ	Valeur du champ
ID de l'exigence	785
Nom	Manuel de Dev & Intégration
Etat exigence	9
Description	

1.1.2.2.1 Manuel DEV/INT : Modèle Manuel DEV/INT

Étiquette du champ	Valeur du champ
ID de l'exigence	1647
Nom	Manuel DEV/INT : Modèle Manuel DEV/INT
Etat exigence	10
Description	Le Manuel DEV/INT doit suivre le modèle et l'outillage défini dans le projet git : http://gitlab.altair.recouv/sded/coeur-de-metier-recouvrement/controle-et-recouvrement-amiable-force-parametres/raf/raf-indus/raf-dacstack

1.1.2.2.2 Manuel DEV/INT - Generation Fichier Word

Étiquette du champ	Valeur du champ
ID de l'exigence	1649
Nom	Manuel DEV/INT - Generation Fichier Word
Etat exigence	8
Description	

Il sera joint au manuel DEV/INT un script powershell de génération du manuel DEV/INT (docx.ps1, cf PJ) qui devra figurer au même niveau que les fichiers markdown. après lancement, le manuel DEV/INT au format docx est correctement générée.

1.1.2.2.3 Manuel DEV/INT : 00 - Header.md

Étiquette du champ	Valeur du champ
ID de l'exigence	1648
Nom	Manuel DEV/INT : 00 - Header.md
Etat exigence	9
Description	Le template du manuel DEV/INT comprend un fichier "00 - header.md" qui doit être rempli sans en changer la structure pour ne pas casser la génération de document word.

1.1.2.2.4 Manuel DEV/INT - 01 - Preamble.md - Objet du document

Étiquette du champ	Valeur du champ
ID de l'exigence	1650
Nom	Manuel DEV/INT - 01 - Preamble.md - Objet du document
Etat exigence	8
Description	Dans le manuel DEV/INT, fichier markdown "01 - Preamble.md", le paragraphe "Objet du document" présentera de façon succincte le module concerné.

1.1.2.2.5 Manuel DEV/INT - 01 - Preamble.md - Documents Applicables

Étiquette du champ	Valeur du champ
ID de l'exigence	1651
Nom	Manuel DEV/INT - 01 - Preamble.md - Documents Applicables
Etat exigence	9
Description	Dans le manuel DEV/INT, fichier markdown "01 - Preamble.md", le paragraphe "Documents Applicables" devra reporter l'ensemble des documents de référence à la rédaction du STD. Cela inclus le STG, le DAT ainsi que le SFD et éventuellement les STD des services consommés (APIs, Messages).

1.1.2.2.6 Développement

Étiquette du champ	Valeur du champ
ID de l'exigence	786
Nom	Développement

Etat exigence	45
Description	
Ceci est un test d'injection d'une exigence	

1.1.2.2.6.1 Manuel DEV/INT - 02 - Developpement.md - Starter Kit FullStack

Étiquette du champ	Valeur du champ
ID de l'exigence	1653
Nom	Manuel DEV/INT - 02 - Developpement.md - Starter Kit FullStack
Etat exigence	14
Description	
Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", le paragraphe "StarterKit FullStack" devra indiquer le cas échéant à partir de quel Starter Kit (Back-End SpringBoot, Front End SPA Angular, ...) les développements ont été réalisés (ou indiquer pourquoi aucun starter kit n'a été utilisé pour le développement)	

1.1.2.2.6.2 Manuel DEV/INT - 02 - Developpement.md - Outils de développement

Étiquette du champ	Valeur du champ
ID de l'exigence	1654
Nom	Manuel DEV/INT - 02 - Developpement.md - Outils de développement
Etat exigence	12
Description	
Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", le paragraphe "Outils de développement" devra indiquer pour chaque outil nécessaire au développement :	
• Les informations concernant l'outil (version, lien de téléchargement, outil local ou partagé,...) • La procédure d'installation (outil local, ex : maven) ou de paramétrage (outil partagé, ex : nexus) de l'outil	

1.1.2.2.6.3 Manuel DEV/INT - 02 - Developpement.md - Stockage et récupération des sources

Étiquette du champ	Valeur du champ
ID de l'exigence	1655
Nom	Manuel DEV/INT - 02 - Developpement.md - Stockage et récupération des sources
Etat exigence	16
Description	
Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", le paragraphe "Construction" devra contenir un sous-paragraphe de stockage et de récupération des sources qui listera l'ensemble des projets GIT nécessaires au développement du module. Le paragraphe devra en outre la procédure pour récupérer les sources des projets.	

1.1.2.2.6.4 Manuel DEV/INT - 02 - Developpement.md - Workflow de développement

Étiquette du champ	Valeur du champ
ID de l'exigence	787
Nom	Manuel DEV/INT - 02 - Developpement.md - Workflow de développement

Etat exigence	32
Description	Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", le paragraphe "Workflow de développement" devra préciser les bonnes pratiques de gestion de configurations devant être suivies par le développeur. Il doit contenir à minima les informations suivantes : - Les règles de nommage des branches - Les règles d'écriture des messages de commit - Le cycle de vie des branches - Le lien vers la norme ACOSS applicable

1.1.2.2.6.5 Manuel DEV/INT - 02 - Developpement.md - Configuration applicative

Étiquette du champ	Valeur du champ
ID de l'exigence	1656
Nom	Manuel DEV/INT - 02 - Developpement.md - Configuration applicative
Etat exigence	13
Description	Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", le paragraphe "Construction" devra contenir un sous-paragraphe de configuration applicative qui listera l'ensemble fichiers de configuration de l'application tout en précisant l'emplacement de ces fichiers de configuration.

1.1.2.2.6.6 Manuel DEV/INT - 02 - Developpement.md - Bouchonnage des applications contributrices

Étiquette du champ	Valeur du champ
ID de l'exigence	1660
Nom	Manuel DEV/INT - 02 - Developpement.md - Bouchonnage des applications contributrices
Etat exigence	14
Description	Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", dans le cas de bouchonnage d'application contributrice (consommation d'API, consommation de message, consommation de fichier), le paragraphe "Bouchonnage des applications contributrices" devra contenir : • un schéma d'interaction entre les différents composants internes et externes au système avec sur chaque flèche représentant une interface, la référence à la procédure de bouchonnage à réaliser. • pour chaque application contributrice bouchonnée, indiquer dans un sous-paragraphe le mode de bouchonnage et/ou décrire le ou les bouchon(s) réalisé(s).

1.1.2.2.6.7 Manuel DEV/INT - 02 - Developpement.md - Compilation

Étiquette du champ	Valeur du champ
ID de l'exigence	1657
Nom	Manuel DEV/INT - 02 - Developpement.md - Compilation
Etat exigence	14
Description	

Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", le paragraphe "Construction" devra contenir un sous-paragraphe concernant la compilation des sources des projets. Ce paragraphe précisera les paramètres de compilation, les étapes de compilation et en particulier les éventuelles étapes de génération de code.

1.1.2.2.6.8 Manuel DEV/INT - 02 - Developpement.md - Exécution sur le poste de développement

Étiquette du champ	Valeur du champ
ID de l'exigence	1658
Nom	Manuel DEV/INT - 02 - Developpement.md - Exécution sur le poste de développement
Etat exigence	16
Description	Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", le paragraphe "Exécution sur le poste de développement" devra préciser comment exécuter chaque projet sur le poste de développement. Il devra notamment préciser - la procédure de lancement de module/composant sur le poste de développement. - Dans le cas de dépendances, l'ordre d'exécution des composants. - Dans le cas d'appels d'API ou de consommation de messages ou de fichiers externes, s'il y a usage de bouchons et les lister.

1.1.2.2.6.9 Manuel DEV/INT - 02 - Developpement.md - Exécution sur le poste de développement - Mode debug

Étiquette du champ	Valeur du champ
ID de l'exigence	1659
Nom	Manuel DEV/INT - 02 - Developpement.md - Exécution sur le poste de développement - Mode debug
Etat exigence	12
Description	Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", le paragraphe "Exécution sur le poste de développement" devra préciser comment exécuter chaque projet sur le poste de développement en mode debug. Il devra notamment préciser - la procédure de lancement de module/composant sur le poste de développement en mode debug. - Dans le cas de dépendances, l'ordre d'exécution des composants. - Dans le cas d'appels d'API ou de consommation de messages ou de fichiers externes, s'il y a usage de bouchons et les lister. - La modification du niveau de log.

1.1.2.2.6.10 Manuel DEV/INT - 02 - Developpement.md - Tests Unitaires

Étiquette du champ	Valeur du champ
ID de l'exigence	1661
Nom	Manuel DEV/INT - 02 - Developpement.md - Tests Unitaires
Etat exigence	11
Description	

Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", le paragraphe "Test" devra contenir un sous-paragraphe concernant exécution des tests unitaires.

1.1.2.2.6.11 Manuel DEV/INT - 02 - Developpement.md - Tests des API avec Postman

Étiquette du champ	Valeur du champ
ID de l'exigence	1662
Nom	Manuel DEV/INT - 02 - Developpement.md - Tests des API avec Postman
Etat exigence	13
Description	Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", dans le cas où le modul expose des API, le paragraphe "Test" devra contenir un sous-paragraphe indiquant les liste des API testées (incluant les verbes http), les paramètres d'entrée s'il y a lieu (incluant les header nécessaires et les payloads si besoin) et les valeurs de retour attendues (codes http, corps de la réponse)

1.1.2.2.6.12 Manuel DEV/INT - 02 - Developpement.md - Tests des messages KAFKA avec AKHQ

Étiquette du champ	Valeur du champ
ID de l'exigence	1663
Nom	Manuel DEV/INT - 02 - Developpement.md - Tests des messages KAFKA avec AKHQ
Etat exigence	13
Description	Dans le manuel DEV/INT, fichier markdown "02 - Developpement.md", dans le cas où le module consomme des messages KAFKA, le paragraphe "Test" devra contenir un sous-paragraphe indiquant la procédure de push des messages sur le bus à l'aide de l'outil AKHQ (compte AKHQ, topic, schéma, structure du message,...)

1.1.2.2.7 Intégration

Étiquette du champ	Valeur du champ
ID de l'exigence	796
Nom	Intégration
Etat exigence	40
Description	

1.1.2.2.7.1 Manuel DEV/INT - 03 - Integration.md - Rappels du modèle de branche UCN

Étiquette du champ	Valeur du champ
ID de l'exigence	1666
Nom	Manuel DEV/INT - 03 - Integration.md - Rappels du modèle de branche UCN
Etat exigence	15
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Rappels du modèle de branche UCN" devra rappeler le modèle de branche UCN

1.1.2.2.7.2 Manuel DEV/INT - 03 - Integration.md - Chaines de construction

Étiquette du champ	Valeur du champ
ID de l'exigence	1667
Nom	Manuel DEV/INT - 03 - Integration.md - Chaines de construction
Etat exigence	11
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Chaines de construction" devra contenir dans un sous paragraphe pour chaque composant les informations suivantes: • nom du projet/composant • URL dépôt GIT • Chemin du JenkinsFile • URL du pipeline Jenkins • Description • Type : multibranche • etapes de construction • Mode de déclenchement • Artefacts produits • "Artefacts construits Maven" contenant les informations suivantes: ▪ GroupId • ArtifactId • Classifier • Version • Descriptif" • "Artefacts construits NPM" contenant les informations suivantes: • Nom • "Artefacts construits Image Docker" contenant les informations suivantes: • Nom • "Artefacts construits Chart Helm" contenant les informations suivantes: • Nom

1.1.2.2.7.3 Manuel DEV/INT - 03 - Integration.md - Mesures de la qualité

Étiquette du champ	Valeur du champ
ID de l'exigence	1668
Nom	Manuel DEV/INT - 03 - Integration.md - Mesures de la qualité
Etat exigence	10
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Mesures de la qualité" devra présenter pour chaque composant concerné : • les choix de paramétrage Sonar (sonar-project.properties), en justifiant les exclusions. • l'url SonarQube du composant • Les accès aux rapports des vulnérabilité trivy

1.1.2.2.7.4 Manuel DEV/INT - 03 - Integration.md - Deploiement

Étiquette du champ	Valeur du champ
ID de l'exigence	1669
Nom	Manuel DEV/INT - 03 - Integration.md - Deploiement
Etat exigence	9
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Déploiement" devra présenter pour chaque composant concerné la procédure de déploiement selon la plateforme cible • HAWAI • PFS • SX5

1.1.2.2.7.5 Manuel DEV/INT - 03 - Integration.md - Deploiement Plateforme HAWAI

Étiquette du champ	Valeur du champ
ID de l'exigence	1670
Nom	Manuel DEV/INT - 03 - Integration.md - Deploiement Plateforme HAWAI
Etat exigence	14
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Déploiement" devra présenter dans un sous-paragraphe pour chaque composant déployé sur plateforme HAWAI et de manière détaillée la procédure d'installation. Le paragraphe devra en particulier être conforme quant à la convention de nommage des lots avec une attention particulière sur les propriété HAWAI. Le paragraphe devra aussi préciser les variables d'environnement par environnement cible : • Les fichiers de configurations utilisés • Les paramètres (descriptif, valeurs possibles) • Ou et comment modifier les valeurs

1.1.2.2.7.6 Manuel DEV/INT - 03 - Integration.md - Deploiement Plateforme SX5

Étiquette du champ	Valeur du champ
ID de l'exigence	1671
Nom	Manuel DEV/INT - 03 - Integration.md - Deploiement Plateforme SX5
Etat exigence	12
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Déploiement" devra présenter dans un sous-paragraphe pour chaque composant déployé sur plateforme SX5 et de manière détaillée la procédure d'installation. Le paragraphe devra aussi préciser les variables d'environnement par environnement cible : • Les fichiers de configurations utilisés • Les paramètres (descriptif, valeurs possibles) • Ou et comment modifier les valeurs

1.1.2.2.7.7 Manuel DEV/INT - 03 - Integration.md - Deploiement Plateforme PFS

Étiquette du champ	Valeur du champ
ID de l'exigence	1672
Nom	Manuel DEV/INT - 03 - Integration.md - Deploiement Plateforme PFS
Etat exigence	11
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Déploiement" devra présenter dans un sous-paragraphe pour chaque composant déployé sur plateforme PFS et de manière détaillée la procédure d'installation. Le paragraphe devra aussi préciser par environnement cible Ou et comment modifier les variables d'environnement.

1.1.2.2.7.8 Manuel DEV/INT - 03 - Integration.md - Deploiement des Bundles Archimed

Étiquette du champ	Valeur du champ
ID de l'exigence	1673
Nom	Manuel DEV/INT - 03 - Integration.md - Deploiement des Bundles Archimed
Etat exigence	9
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Déploiement" devra présenter dans un sous-paragraphe pour chaque composant exposant des services la procédure de déploiement du Bundle Archimed Le paragraphe devra aussi préciser le contenu du Bundle Archimed : • Contrat d'API • HealthCheck • Fichier de règle • ...

1.1.2.2.7.9 Manuel DEV/INT - 03 - Integration.md - Tests d'intégration - accès aux services

Étiquette du champ	Valeur du champ
ID de l'exigence	1675
Nom	Manuel DEV/INT - 03 - Integration.md - Tests d'intégration - accès aux services
Etat exigence	9
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Tests d'intégration" devra présenter dans un sous-paragraphe listant l'ensemble des accès aux différents services du module • Url des Frontend ou des API via Archimed • Url des Frontend ou des API sans Archimed (Hors Prod) • Accès aux Buckets S3 le cas échéant • Accès aux topics KAFKA via AKHQ le cas échéant

- Accès aux serveurs pour lancement de traitements batch le cas échéant

1.1.2.2.7.10 Manuel DEV/INT - 03 - Integration.md - Tests d'intégration - exécution des tests d'intégration

Étiquette du champ	Valeur du champ
ID de l'exigence	1676
Nom	Manuel DEV/INT - 03 - Integration.md - Tests d'intégration - exécution des tests d'intégration
Etat exigence	9
Description	Dans le manuel DEV/INT, fichier markdown "03 - Integration.md", le paragraphe "Tests d'intégration" devra présenter dans un sous-paragraphe listant la procédure d'exécution des tests d'intégration

1.1.2.2.7.11 Manuel DevInt/SNV2 - Chapitre traitant de l'exploitation sur SNV2

Étiquette du champ	Valeur du champ
ID de l'exigence	805
Nom	Manuel DevInt/SNV2 - Chapitre traitant de l'exploitation sur SNV2
Etat exigence	44
Description	Le manuel DevInt doit contenir un chapitre faisant référence à la documentation d'exploitation du produit informatique. Cette dernière décrit la configuration du produit informatique sur socle SX et la manière de l'exploiter. Voir le modèle de document de l'exploitation des containers dans la STG

1.1.2.2.7.12 Manuel DevInt/HAWAI - Chapitre traitant de l'exploitation sur Hawai

Étiquette du champ	Valeur du champ
ID de l'exigence	806
Nom	Manuel DevInt/HAWAI - Chapitre traitant de l'exploitation sur Hawai
Etat exigence	41
Description	Le manuel DevInt doit contenir un chapitre faisant référence à la documentation d'exploitation du Système d'information déployé sur une plateforme HAWAI. Cette dernière décrit le fonctionnement de la plateforme et la manière de l'exploiter. Voir le modèle de document de l'exploitation des plateformes HAWAI dans la STG.

1.1.2.2.7.13 Manuel DevInt/PFS - Chapitre traitant de l'exploitation sur PFS

Étiquette du champ	Valeur du champ
ID de l'exigence	807
Nom	Manuel DevInt/PFS - Chapitre traitant de l'exploitation sur PFS
Etat exigence	38

Description
Le manuel Dev/Int doit contenir un chapitre faisant référence à la documentation d'exploitation du produit informatique sur la plateforme PFS. Cette dernière doit notamment décrire les actions d'exploitation à réaliser en fonction des alertes remontées par le produit informatique.

1.1.2.3 Spécifications Techniques Détaillées

Étiquette du champ	Valeur du champ
ID de l'exigence	809
Nom	Spécifications Techniques Détaillées
Etat exigence	65
Description	

1.1.2.3.1 STD - Modèle STD DACStack

Étiquette du champ	Valeur du champ
ID de l'exigence	1453
Nom	STD - Modèle STD DACStack
Etat exigence	62
Description	La Spécification Technique Détaillée doit suivre le modèle et l'outillage défini dans le projet git : http://gitlab.altair.recouv/sded/coeur-de-metier-recouvrement/controle-et-recouvrement-amiable-force-parametres/raf/raf-indus/raf-dacstack

1.1.2.3.2 STD - Génération document word

Étiquette du champ	Valeur du champ
ID de l'exigence	1457
Nom	STD - Génération document word
Etat exigence	62
Description	Il sera joint à la STD un script powershell de génération de la STD (docx.ps1, cf PJ) qui devra figurer au même niveau que les fichiers asciidoc. après lancement, la STD au format docx est correctement générée.

1.1.2.3.3 STD - 00 - Header

Étiquette du champ	Valeur du champ
ID de l'exigence	1424
Nom	STD - 00 - Header
Etat exigence	61
Description	

Le template de la STD comprend un fichier "00 - header" qui doit être rempli sans en changer la structure pour ne pas casser la génération de document word.

1.1.2.3.4 STD - 01 - Preamble - Objet du document - Objet du document

Étiquette du champ	Valeur du champ
ID de l'exigence	1426
Nom	STD - 01 - Preamble - Objet du document - Objet du document
Etat exigence	64
Description	Dans la STD, fichier "01 - Preamble", le paragraphe "Objet du document" présentera de façon succincte le module concerné.

1.1.2.3.5 STD - 01 - Preamble - Documents Applicables

Étiquette du champ	Valeur du champ
ID de l'exigence	1427
Nom	STD - 01 - Preamble - Documents Applicables
Etat exigence	60
Description	Dans la STD, fichier "01 - Preamble", le paragraphe "Documents Applicables" devra reporter l'ensemble des documents de référence à la rédaction du STD. Cela inclus le STG, le DAT ainsi que le SFD et éventuellement les STD des services consommés (APIs, Messages).

1.1.2.3.6 STD - 02 - Module - Récapitulatif - Contexte

Étiquette du champ	Valeur du champ
ID de l'exigence	1428
Nom	STD - 02 - Module - Récapitulatif - Contexte
Etat exigence	59
Description	Dans la STD, fichier "02 - Module", le tableau du paragraphe "Contexte" devra être dûment renseigné.

1.1.2.3.7 STD - 02 - Module - Récapitulatif - Périmètre de la solution

Étiquette du champ	Valeur du champ
ID de l'exigence	1429
Nom	STD - 02 - Module - Récapitulatif - Périmètre de la solution
Etat exigence	59
Description	Dans la STD, fichier "02 - Module", le paragraphe "Périmètre de la solution" devra situer le module dans le contexte global du projet PXX + de l'itération en cours. Il devra en particulier contenir un schéma donnant une vue applicative de Projet

1.1.2.3.8 STD - 03 - Rappel de spécifications générales

Étiquette du champ	Valeur du champ
ID de l'exigence	1430
Nom	STD - 03 - Rappel de spécifications générales
Etat exigence	62
Description	Dans la STD, fichier "03 - Rappel de spécifications generales", Rappeler en introduction le schéma d'architecture applicative du DAT et (re) préciser le contexte du module

1.1.2.3.9 STD - 03 - Rappel de spécifications générales - Organisation des sources - Librairies - Internes

Étiquette du champ	Valeur du champ
ID de l'exigence	1431
Nom	STD - 03 - Rappel de spécifications générales - Organisation des sources - Librairies - Internes
Etat exigence	64
Description	Dans la STD, fichier "03 - Rappel de spécifications generales", le tableau du paragraphe "Librairies internes" devra lister l'ensemble des librairies internes à UCN utilisées par ce module

1.1.2.3.10 STD - 03 - Rappel de spécifications générales - Organisation des sources - Librairies - Externes

Étiquette du champ	Valeur du champ
ID de l'exigence	1432
Nom	STD - 03 - Rappel de spécifications générales - Organisation des sources - Librairies - Externes
Etat exigence	61
Description	Dans la STD, fichier "03 - Rappel de spécifications generales", le tableau du paragraphe "Librairies externes" devra lister l'ensemble des librairies du marché utilisées par ce module.

1.1.2.3.11 STD - 03 - Rappel de spécifications générales - Organisation des sources - Logiciels sur étagère

Étiquette du champ	Valeur du champ
ID de l'exigence	1433
Nom	STD - 03 - Rappel de spécifications générales - Organisation des sources - Logiciels sur étagère
Etat exigence	60

Description
Dans la STD, fichier "03 - Rappel de spécifications générales", le paragraphe "Logiciels sur étagère" devra décrire pour chaque logiciel sur étagère : • le nom • la version • une description • une liste de recommandations pour une configuration et une utilisation sécurisée du logiciel dans l'environnement UCN. • une justification de l'utilisation du COTS devant aborder les points suivants: ○ une présentation succincte du COTS et des solutions alternatives ○ les avantages/inconvénients • les impacts sur la conception ○ les licences

1.1.2.3.12 STD - 03 - Rappel de spécifications générales - Organisation des sources - Logiciels sur étagère (licences)

Étiquette du champ	Valeur du champ
ID de l'exigence	1455
Nom	STD - 03 - Rappel de spécifications générales - Organisation des sources - Logiciels sur étagère (licences)
Etat exigence	61
Description	
	La STD doit contenir un chapitre décrivant les licences des COTS utilisées. Pour chaque licence les points suivants doivent être présents: • Un descriptif • L'aspect contaminant ou non • Les obligations inhérentes à l'utilisation des COTS sous cette licence (ex: présence d'un fichier dans les sources ...) • L'url vers le site de référence de la licence

1.1.2.3.13 STD - 05 - Spécifications du module - Vue d'ensemble - Vue des composants et des services du module

Étiquette du champ	Valeur du champ
ID de l'exigence	1434
Nom	STD - 05 - Spécifications du module - Vue d'ensemble - Vue des composants et des services du module
Etat exigence	61
Description	
	Dans la STD, fichier "05 - Spécifications du module", le paragraphe "Vue des fonctionnalités portées par le module" devra introduire le schéma de la vue applicative détaillée.

1.1.2.3.14 STD - 05 - Spécifications du module - Vue d'ensemble - Vue des fonctionnalités portées par le module

Étiquette du champ	Valeur du champ
ID de l'exigence	1435
Nom	STD - 05 - Spécifications du module - Vue d'ensemble - Vue des fonctionnalités portées par le module
Etat exigence	64
Description	Dans la STD, fichier "05 - Spécifications du module", le paragraphe "Vue des fonctionnalités portées par le module" devra introduire un schéma de la vue des services et des fonctions de gestion et en décrire pour chacun/chacune leur rôle.

1.1.2.3.15 STD - 05 - Spécifications du module - Vue d'ensemble - Vue des ressources portées par le module

Étiquette du champ	Valeur du champ
ID de l'exigence	3220
Nom	STD - 05 - Spécifications du module - Vue d'ensemble - Vue des ressources portées par le module
Etat exigence	11
Description	Dans la STD, fichier "05 - Spécifications du module", le paragraphe "Vue des ressources portées par le module" devra introduire un schéma de ressources/sous-ressources métiers portées par le module et en décrire pour chacune son niveau (ressource/sous-ressources), son rôle, ses liens, sa logique de cycle de vie (création, mise à jour, destruction), ...

1.1.2.3.16 STD - 05 - Spécifications du module - Détail des composants

Étiquette du champ	Valeur du champ
ID de l'exigence	1436
Nom	STD - 05 - Spécifications du module - Détail des composants
Etat exigence	62
Description	Dans la STD, fichier "05 - Spécifications du module", le paragraphe "Détail des composants" devra lister les composants de ce module (Composant Archimed, Composant API, Composant BD, Composant Kafka)

1.1.2.3.17 STD - 05 - Spécifications du module - Détail des composants - Composant Bundle Archimed

Étiquette du champ	Valeur du champ
ID de l'exigence	1437
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant Bundle Archimed
Etat exigence	64
Description	

Au cas où le module expose des services "à l'extérieur" (donc via Archimed), dans la STD, fichier "05 - Spécifications du module", le tableau du paragraphe "Composant Archimed" devra être dûment rempli, avec en particulier :

- Nom : Contrat d'interface XXX
- Description :
- Détail Techniques :
- Paquet/Dépôt :

1.1.2.3.18 STD - 05 - Spécifications du module - Détail des composants - Composant Base de Données

Étiquette du champ	Valeur du champ
ID de l'exigence	1438
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant Base de Données
Etat exigence	67
Description	Au cas où le module persiste des données, dans la STD, fichier "05 - Spécifications du module", le tableau du paragraphe "Composant Base de données" devra être dûment rempli, avec en particulier : • Nom : Base de données XXX • Namespace : • Database : • Schéma : • Port : 5432 Le paragraphe devra aussi contenir le diagramme du MPD. En pièce jointe de la STD, un fichier SQL de création du data model pourra être fourni.

1.1.2.3.19 STD - 05 - Spécifications du module - Détail des composants - Composant Topic Kafka

Étiquette du champ	Valeur du champ
ID de l'exigence	1439
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant Topic Kafka
Etat exigence	67
Description	Au cas où le module émet ou consomme des messages KAFKA, dans la STD, fichier "05 - Spécifications du module", le paragraphe "Composant Topic KAFKA" devra décrire : • Les messages. Chaque message doit être spécifié dans un fichier AVRO • Les topics ○ partitions ○ la nature (inter-domaine, intradomaine ...) • La stratégie de consommation ○ Buffer

- Agrégation
- Capacité de consommation
- Gestion des erreurs et commits

N.B. : Tout écart de structure d'un message par rapport au format attendu doit être décrit et justifié.

1.1.2.3.20 STD - 05 - Spécifications du module - Détail des composants - Composant API - Diagramme de classe

Étiquette du champ	Valeur du champ
ID de l'exigence	1440
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant API - Diagramme de classe
Etat exigence	65
Description	Dans la STD, fichier "05 - Spécifications du module", le paragraphe "Diagramme de classe" devra contenir un diagramme qui devra préciser les classes DTO et les interfaces de gestion des DTO. Dans le cas d'entités à état, la STD devra inclure le diagramme d'état de ces entités.

1.1.2.3.21 STD - 05 - Spécifications du module - Détail des composants - Composant API - Enumérations

Étiquette du champ	Valeur du champ
ID de l'exigence	1441
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant API - Enumérations
Etat exigence	65
Description	Au cas où l'API utilise des données de type énuméré, dans la STD, fichier "05 - Spécifications du module", le paragraphe "Enumérations" devra décrire l'ensemble de ces types avec la liste exhaustive des valeurs.

1.1.2.3.22 STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Mapping Table-Entité

Étiquette du champ	Valeur du champ
ID de l'exigence	1442
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Mapping Table-Entité
Etat exigence	64
Description	Au cas où l'API persiste des données, dans la STD, fichier "05 - Spécifications du module", le paragraphe "Mapping Table-Entité" devra contenir le mapping Table - Entité pour chaque table du modèle.

1.1.2.3.23 **STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Mapping Entité-DTO**

Étiquette du champ	Valeur du champ
ID de l'exigence	1443
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Mapping Entité-DTO
Etat exigence	64
Description	Au cas où l'API persiste des données, dans la STD, fichier "05 - Spécifications du module", le paragraphe "Mapping Entité-DTO" devra contenir le mapping Entité - DTO pour chaque entité du paragraphe précédent.

1.1.2.3.24 **STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Mapping Message KAFKA-DTO**

Étiquette du champ	Valeur du champ
ID de l'exigence	1444
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Mapping Message KAFKA-DTO
Etat exigence	63
Description	Au cas où l'API persiste des données, dans la STD, fichier "05 - Spécifications du module", le paragraphe "Mapping Message KAFKA - DTO" devra contenir le mapping Message KAFKA - DTO pour chaque Message KAFKA.

1.1.2.3.25 **STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Génération fichier**

Étiquette du champ	Valeur du champ
ID de l'exigence	1445
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant API - Mappings - Génération fichier
Etat exigence	64
Description	Au cas où l'API persiste des données, dans la STD, fichier "05 - Spécifications du module", le paragraphe "Génération fichier" devra décrire le format et le contenu du fichier (plat, csv, ...)

1.1.2.3.26 **STD - 05 - Spécifications du module - Détail des composants - Composant API - Services**

Étiquette du champ	Valeur du champ
ID de l'exigence	1446
Nom	STD - 05 - Spécifications du module - Détail des composants - Composant API - Services
Etat exigence	63
Description	

Dans la STD, fichier "05 - Spécifications du module", le paragraphe "Services" devra contenir la description de l'ensemble des services. Cela contient :

- Le type de service (lecture, création, modification, suppression)
- Les paramètres d'entrée
- Les paramètres de sortie
- Les messages d'erreur

Dans le cas de services complexes, la STD devra contenir un diagramme de séquence.

1.1.2.3.27 STD - 06 - Interfaces - Schéma des échanges

Étiquette du champ	Valeur du champ
ID de l'exigence	1447
Nom	STD - 06 - Interfaces - Schéma des échanges
Etat exigence	61
Description	Dans la STD, fichier "06 - interfaces", le paragraphe "Schéma des échanges" devra include un schéma des échanges du module avec les autres modules

1.1.2.3.28 STD - 06 - Interfaces - Interfaces fournies

Étiquette du champ	Valeur du champ
ID de l'exigence	1448
Nom	STD - 06 - Interfaces - Interfaces fournies
Etat exigence	64
Description	Dans la STD, fichier "06 - interfaces", le paragraphe "Interfaces fournies" devra énumérer la liste des interfaces fournies en précisant : • son uri/url • son type de requête HTTP (GET, POST, PUT, ...) • les paramètres d'entrée • le cas échéant le paramètre de sortie • les codes retour HTTP (200, 400, 500, ...) • Si le service est exposable via Archimed En pièce jointe de la STD, un fichier yaml swagger devra être fourni.

1.1.2.3.29 STD - 06 - Interfaces - Interfaces consommées

Étiquette du champ	Valeur du champ
ID de l'exigence	1449
Nom	STD - 06 - Interfaces - Interfaces consommées
Etat exigence	64
Description	Dans la STD, fichier "06 - interfaces", le paragraphe "Interfaces consommées" devra énumérer la liste des interfaces consommées en précisant : • son namespace • Si le service est appelé via Archimed ou dans le même namespace

- son uri/url
- son type de requête HTTP (GET, POST, PUT, ...)
- les paramètres d'entrée
- le cas échéant le paramètre de sortie
- les codes retour HTTP (200, 400, 500, ...)

En pièce jointe de la STD, un fichier yaml swagger (REST) ou wsdl (SOAP) du service consommé devra être fourni.

1.1.2.3.30 STD - 06 - Interfaces - Bus de messages KAFKA

Étiquette du champ	Valeur du champ
ID de l'exigence	1450
Nom	STD - 06 - Interfaces - Bus de messages KAFKA
Etat exigence	69
Description	Dans la STD, fichier "06 - interfaces", le paragraphe "Bus de messages KAFKA" devra énumérer la liste des messages, en précisant : • Le nom du fichier (format si imposé) • L'url du fichier AVRO • Les composants producteurs et les services émettant le message • Les composants consommateurs et les services consommant le message • Les événements provoquant le déclenchement des messages • Comment les champs sont valorisés et quelles sont les fonctions internes appelées • La criticité des données contenues dans le message (sécurisée/sensible/normal)

1.1.2.3.31 STD - 06 - Interfaces - Traitement de Fichier - Bucket S3

Étiquette du champ	Valeur du champ
ID de l'exigence	1646
Nom	STD - 06 - Interfaces - Traitement de Fichier - Bucket S3
Etat exigence	67
Description	Dans la STD, fichier "06 - interfaces", le paragraphe "Traitement de fichiers - Bucket S3" devra énumérer la liste des fichiers, en précisant : • Le nom du fichier (format si imposé) • La structure du fichier • La périodicité de génération • Le Bucket S3 concerné

1.1.2.3.32 STD - 06 - Interfaces - Supervision

Étiquette du champ	Valeur du champ
ID de l'exigence	1451
Nom	STD - 06 - Interfaces - Supervision

Etat exigence	64
Description	
Dans la STD, fichier "06 - interfaces", le paragraphe "Supervision" devra préciser : • Healthcheck (url, statut, message, ...) • Métriques spécifiques à suivre	

1.1.2.3.33 STD - 07 - Intégration

Étiquette du champ	Valeur du champ
ID de l'exigence	1452
Nom	STD - 07 - Intégration
Etat exigence	63
Description	
La STD doit contenir les informations de base sur les pipelines d'intégration continue implémentés • Version des outils de build • La liste des chaînes de constructions et les étapes les constituants • Les caractéristiques et le but de chaque chaîne de construction • Les séquences de lancement des chaînes de constructions de la génération des composants à leur déploiement.	

1.1.2.4 Dossier d'Exploitation

Étiquette du champ	Valeur du champ
ID de l'exigence	1279
Nom	Dossier d'Exploitation
Etat exigence	9
Description	

1.1.2.4.1 DEX - Modèle DEX

Étiquette du champ	Valeur du champ
ID de l'exigence	1280
Nom	DEX - Modèle DEX
Etat exigence	21
Description	
Le DEX doit être basé sur le modèle doc as code suivant : https://readthedocs.cnp.recouv/docs/modele-et-notice-dex/en/latest/ Les modèle word historiques peuvent servir de base pour remplir certaines sections	

http://gitlab.altair.recouv/sded/coeur-de-metier-recouvrement/controle-et-recouvrement-amiable-force-parametres/raf/raf-referentiel-projet/raf-document-techniques-applicables/-/raw/1.10.0/Documents%20Techniques%20Applicables/Mod%C3%A8les/Dossier-d-Exploitation-MODELE_008.doc
http://gitlab.altair.recouv/sded/coeur-de-metier-recouvrement/controle-et-recouvrement-amiable-force-parametres/raf/raf-referentiel-projet/raf-document-techniques-applicables/-/raw/1.10.0/Documents%20Techniques%20Applicables/Mod%C3%A8les/Dossier-d-Exploitation-NOTICE_008.doc

1.1.2.4.2 DEX - Procédures d'exploitations

Étiquette du champ	Valeur du champ
ID de l'exigence	1281
Nom	DEX - Procédures d'exploitations
Etat exigence	10
Description	
Le DEX doit contenir l'ensemble des procédures d'Exploitation identifiés.	

1.1.2.4.3 DEX - Métriques et supervision

Étiquette du champ	Valeur du champ
ID de l'exigence	1282
Nom	DEX - Métriques et supervision
Etat exigence	13
Description	
Le DEX doit identifier les métriques utiles et comment y accéder via un dashboard ou une api.	

1.1.2.4.4 DEX - Sondes

Étiquette du champ	Valeur du champ
ID de l'exigence	1283
Nom	DEX - Sondes
Etat exigence	10
Description	
Le DEX doit contenir les informations nécessaires à la mise en place de sondes d'alerte par les équipes d'exploitation en décrivant métrique, seuil et les actions à réaliser en cas de dépassement de ces derniers.	

1.1.2.4.5 DEX - Composants Applicatifs

Étiquette du champ	Valeur du champ
ID de l'exigence	1284
Nom	DEX - Composants Applicatifs
Etat exigence	10
Description	
Le DEX doit comprendre une description des composants applicatifs et de leur contexte fonctionnel.	

Il doit aussi décrire les différents flux d'échanges entre composants et vers l'extérieur.

1.1.2.4.6 DEX - Outillage

Étiquette du champ	Valeur du champ
ID de l'exigence	1285
Nom	DEX - Outillage
Etat exigence	10
Description	Le DEX doit comprendre une liste des outils nécessaire à l'exploitation selon les procédures décrites.

1.1.2.4.7 DEX - Tâches planifiées

Étiquette du champ	Valeur du champ
ID de l'exigence	1286
Nom	DEX - Tâches planifiées
Etat exigence	10
Description	Le DEX doit décrire l'ensemble des Tâches planifiées du système afin qu'elle soit installées et maintenues par les exploitants.

1.1.2.5 Documentation générée par GLD

Étiquette du champ	Valeur du champ
ID de l'exigence	1394
Nom	Documentation générée par GLD
Etat exigence	15
Description	

1.1.2.5.1 Livraison/SNV2 - Versioning documentation dans GLD

Étiquette du champ	Valeur du champ
ID de l'exigence	1093
Nom	Livraison/SNV2 - Versioning documentation dans GLD
Etat exigence	21
Description	Lors de la livraison d'un nouvel élément doc1, la version du composant s'incrémente. Cette version doit être obligatoirement reportée dans la fiche de planification à téléverser dans la fiche GLD Pour les composants DOC1, il faut renseigner le champ « Fichiers adélaide » dans l'onglet « Technique » Référence : Documentation GLD (chapitre 6.8. CONSIGNE ADELAIDE)

1.1.2.5.2 Livraison/SNV2 - Déclaration de la modification d'une base de données dans GLD

Étiquette du champ	Valeur du champ
ID de l'exigence	1094
Nom	Livraison/SNV2 - Déclaration de la modification d'une base de données dans GLD
Etat exigence	18
Description	Lors de la création ou modification de la structure de base de données SNV2, la procédure de la documentation GLD suivante doit être réalisée : Référence : Documentation GLD (chapitre 6.5. Table et Fichiers divers)

1.1.2.5.3 Backend/SNV2 - Modification de traitements/services

Étiquette du champ	Valeur du champ
ID de l'exigence	865
Nom	Backend/SNV2 - Modification de traitements/services
Etat exigence	16
Description	Lors de la création ou modification d'un traitement, d'une transaction SAI2 ou d'un service, le développeur doit suivre la procédure de la documentation GLD suivante : Référence : Documentation GLD (chapitre 6.7. TRAITEMENTS, TRANSACTIONS ET SERVICES)

1.1.2.5.4 Livrable - Exigences fonctionnelles du GLD

Étiquette du champ	Valeur du champ
ID de l'exigence	1084
Nom	Livrable - Exigences fonctionnelles du GLD
Etat exigence	18
Description	Dans la documentation GLD (Guichet de Livraison Documentaire) le prestataire doit réaliser les exigences fonctionnelles du SDIT.

1.1.2.5.5 Livrable/SNV2 - Fiche de planification

Étiquette du champ	Valeur du champ
ID de l'exigence	1071
Nom	Livrable/SNV2 - Fiche de planification
Etat exigence	17
Description	La fiche de planification doit suivre le template du fichier référencé dans les documents de référence. L'information « Nombre de ressources » est obligatoire. Les valeurs possibles sont les suivantes : 01 : pour les commandes globales (traitements) 20 : pour les commandes individuelles (produits individuels et commandes signal)

1.1.2.5.6 Livrable/SNV2 - Documentation GLD

Étiquette du champ	Valeur du champ
ID de l'exigence	1072
Nom	Livrable/SNV2 - Documentation GLD
Etat exigence	23
Description	Lors de la livraison d'un container, la documentation GLD (Guichet de Livraison Documentaire) relative aux consignes CNP sur le container est obligatoire. Elle doit être jointe dans l'onglet « Documents ».

1.1.2.6 Doc de lot / Install

Étiquette du champ	Valeur du champ
ID de l'exigence	1799
Nom	Doc de lot / Install
Etat exigence	5
Description	

1.1.2.6.1 Doc de lot - Format

Étiquette du champ	Valeur du champ
ID de l'exigence	1800
Nom	Doc de lot - Format
Etat exigence	12
Description	Lors de la livraison d'un projet, la doc de lot qui l'accompagne doit suivre le format attendu par la SDOPE. Il s'agit d'un fichier yaml permettant la génération d'un site readthedoc. https://gitlab.cnp.recouv/sdope/deploiements/docs-installations/template_doc

1.1.2.6.2 Doc de lot - Granularité

Étiquette du champ	Valeur du champ
ID de l'exigence	2406
Nom	Doc de lot - Granularité
Etat exigence	9
Description	Les doc de lot doivent être découpées par chart helm afin de permettre à bonne prise en compte des changement de values.

1.1.2.7 Changelog

Étiquette du champ	Valeur du champ
ID de l'exigence	1641

Nom	Changelog
Etat exigence	3
Description	

1.1.2.7.1 Changelog

Étiquette du champ	Valeur du champ
ID de l'exigence	1642
Nom	Changelog
Etat exigence	238
Description	Les modifications techniques majeures doivent être listées dans un fichier "Changelog.md" sur les dépôts git correspondants. Les changement suivant doivent en particulier y figurer : • Ajout de variables dans les charts • Ajout d'appels externes • Changements de contrat d'interface

1.1.2.8 DocAsCode

Étiquette du champ	Valeur du champ
ID de l'exigence	1644
Nom	DocAsCode
Etat exigence	15
Description	Les livrables documentaires suivants doivent être réalisés en doc as code sur la base des modèles fournis dans le dépôt suivant : http://gitlab.altair.recouv/sded/coeur-de-metier-recouvrement/controle-et-recouvrement-amiable-force-parametres/raf/raf-indus/raf-dacstack/-/tree/master/src Les documents en doc as code doivent pouvoir être transformé en document Word à l'aide des script inclus dans le projet DACStack sans erreur. En particulier : • Utilisation de ASCIIDoc • Utilisation de diagrammes as code • Bonne génération de la page de garde (Bug possible en cas de mauvais respect de la structure du fichier Header.md) • Présence des images dans le document final (Bug possible lié au chemins de fichiers images dans le MD) • Niveaux de titre • Listes à puce (bug possible en cas de saut de ligne manquant avant la liste dans le fichier MD)

1.1.3 Contraintes de Conception

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	1637
Nom	Contraintes de Conception
Etat exigence	13
Description	

1.1.3.1 Conception - Identification des ressources par UUID

Étiquette du champ	Valeur du champ
ID de l'exigence	1638
Nom	Conception - Identification des ressources par UUID
Etat exigence	13
Description	Toute instance d'entité est identifiée par un UUID. On peut sortir de la règle des UUID si (and): • L'énumération est embarquée dans du code • Le nombre d'éléments de l'énumération ne dépasse pas 40 Dans ce cas, l'identifiant de chaque élément peut être un identifiant lisible au format String. Letype de la colonne BDD sera un type ENUM pour permettre la consistance des données dans la BDD sans créer une table dédié. Exemple: CREATE TYPE mood AS ENUM ('sad', 'ok', 'happy'); cf: https://www.postgresql.org/docs/current/datatype-enum.html Attention: ce type BDD doit être crée par script (liquibase).

1.1.3.2 Conception - Suppression logique

Étiquette du champ	Valeur du champ
ID de l'exigence	1639
Nom	Conception - Suppression logique
Etat exigence	11
Description	La suppression logique doit être vue comme une suppression du point de vue de l'utilisateur de l'API: • I.E. s'il effectue une requête GET sur l'élément supprimé, alors l'API lui indique que la ressource n'existe pas. • Les propriétés de l'entité (persistées en BDD) et qui permettent la gestion de cette suppression, ne sont pas disponible sur les DTO requête et réponse. La restauration d'une ressource supprimée logiquement pourra se faire ultérieurement sous la forme d'une fonction d'API spécifique.

1.1.3.3 Conception - API

Étiquette du champ	Valeur du champ
ID de l'exigence	1640
Nom	Conception - API
Etat exigence	10
Description	Si le contrat d'interface d'une application inclu des endpoints pour créer ou rechercher une ressource alors il doit aussi inclure un endpoint d'appel de type getByID pour récupérer une ressource unique.

1.1.3.4 Conception - Mutualisation des sources Agent et API

Étiquette du champ	Valeur du champ
ID de l'exigence	2554
Nom	Conception - Mutualisation des sources Agent et API
Etat exigence	7
Description	Les composant agent et api d'un même module doivent être implémentés dans le même projet java. Un paramètre au lancement de l'application Spring doit permettre d'activer la partie kafka. Le projet doit générer 2 images dockers. Une pour chaque mode de fonctionnement de l'application. Le chart mettra en place 2 deployments utilisant chacun une des images.

1.1.3.5 Conception - Scallabilité

Étiquette du champ	Valeur du champ
ID de l'exigence	3211
Nom	Conception - Scallabilité
Etat exigence	4
Description	Les composants de la solution doivent pouvoir être déployés plusieurs fois dans le cadre d'un scaling horizontal. Les conception proposées pour la PFSv2 doivent aussi tenir compte du fonctionnement en bi-cluster.

1.1.3.6 Conception - Mutualisation

Étiquette du champ	Valeur du champ
ID de l'exigence	3215
Nom	Conception - Mutualisation
Etat exigence	4
Description	Chaque fois que c'est possible des librairie java doivent être créées ou mise à jour pour pour porter les source qui sont utilisées par plusieurs applications. En particulier dans les cas suivants : Utilisation de la même base de données

- Contrat d'interface ou lib cliente
- Ensembles de classes utilitaires
- odata

1.1.3.7 Conception - NullPointer

Étiquette du champ	Valeur du champ
ID de l'exigence	3216
Nom	Conception - NullPointer
Etat exigence	4
Description	L'application ne doit pas générer de NullPointerException. Les variables doivent être testée préalablement.

1.1.3.8 Conception - Verrous optimistes

Étiquette du champ	Valeur du champ
ID de l'exigence	3234
Nom	Conception - Verrous optimistes
Etat exigence	4
Description	Plusieurs cas d'usage peuvent aboutir à des concurrences d'accès en écriture sur les données (Deux utilisateurs peuvent être amenés à effectuer une modification sur une même instance, un traitement automatique peut être amené à tenter d'effectuer une modification sur une instance en même temps qu'un utilisateur via l'IHM.) Afin de garantir que le deuxième traitement ne vienne écraser le résultat du premier, et laisser le système dans un état potentiellement incohérent, toute modification effectuée par un traitement devra préalablement s'assurer que les données modifiées sont celles qui ont servi de référence au traitement (lock optimiste). Les verrous optimistes lorsque nécessaires seront implémentés avec l'annotation @version JPA.

1.1.4 Contraintes de Développement

Étiquette du champ	Valeur du champ
ID de l'exigence	824
Nom	Contraintes de Développement
Etat exigence	13
Description	

1.1.4.1 Documents applicables

Étiquette du champ	Valeur du champ
ID de l'exigence	1288
Nom	Documents applicables

Etat exigence	56
Description	
Ceci est un test d'injection d'une exigence	

1.1.4.2 Composants applicatifs

Étiquette du champ	Valeur du champ
ID de l'exigence	862
Nom	Composants applicatifs
Etat exigence	10
Description	

1.1.4.2.1 Backend

Étiquette du champ	Valeur du champ
ID de l'exigence	863
Nom	Backend
Etat exigence	7
Description	

1.1.4.2.1.1 Généralités

Étiquette du champ	Valeur du champ
ID de l'exigence	864
Nom	Généralités
Etat exigence	19
Description	
Ceci est un test d'injection d'une exigence	

1.1.4.2.1.1.1 COTS - Version du Jdk

Étiquette du champ	Valeur du champ
ID de l'exigence	868
Nom	COTS - Version du Jdk
Etat exigence	50
Description	
La version jdk doit être 11 ou plus. Le fabricant s'engage à ne pas utiliser Oracle Java SE mais uniquement les packages proposés par OpenJDK.	

1.1.4.2.1.1.2 Backend - Utilisation des Enum

Étiquette du champ	Valeur du champ
ID de l'exigence	1900
Nom	Backend - Utilisation des Enum
Etat exigence	13

Description
L'utilisation des Enum Java classique doit être réservé au cas où les valeurs n'évoluent absolument pas. Le plus souvent il sera préférable d'utiliser les configurations YAML pour initialiser les données et permettre une surcharge à partir des values.yaml (helm) Les listes identifiées comme plus changeantes devront être intégrées dans des bases de données référentielles

1.1.4.2.1.1.3 Backend - Utilisation des Bucket S3

Étiquette du champ	Valeur du champ
ID de l'exigence	1904
Nom	Backend - Utilisation des Bucket S3
Etat exigence	9
Description	
	L'utilisation des bucket S3 doit suivre les recommandations décrites dans les normes. En particulier la création des buckets doit se faire via le subchart "bucket" en respectant la typologie de bucket correspondant au cas d'usage et la nomenclature demandée.

1.1.4.2.1.1.4 Backend - Stockage des fichiers

Étiquette du champ	Valeur du champ
ID de l'exigence	2404
Nom	Backend - Stockage des fichiers
Etat exigence	9
Description	
	A défaut d'exigences d'architecture spécifique, le stockage de fichiers doit se faire sur la plateforme S3

1.1.4.2.1.1.5 Backend - Pom parent

Étiquette du champ	Valeur du champ
ID de l'exigence	3214
Nom	Backend - Pom parent
Etat exigence	12
Description	
	Les applications java doivent se baser sur le pom parent RAF défini ici : https://gitlab.altair.recouv/sded/coeur-de-metier-recouvrement/contrôle-et-recouvrement-amiable-force-paramètres/raf/raf-transverse/raf-parent en cas de mise à jour de fullstack ce dernier doit être mis à jour pour récupérer les bonnes versions dans les applications

1.1.4.2.1.2 API

Étiquette du champ	Valeur du champ
ID de l'exigence	869
Nom	API
Etat exigence	15
Description	

1.1.4.2.1.2.1 API Backend - API Rest

Étiquette du champ	Valeur du champ
ID de l'exigence	870
Nom	API Backend - API Rest
Etat exigence	32
Description	Les APIs fournies par les services doivent utiliser le protocole REST. Les normes adoptées pour les contrats REST sont la spécification Open API 3.0 Les règles interne à suivre pour les api sont décrites dans la norme fullstack (voir doc applicables)

1.1.4.2.1.2.2 API Backend - Compatibilité des API avec OData

Étiquette du champ	Valeur du champ
ID de l'exigence	871
Nom	API Backend - Compatibilité des API avec OData
Etat exigence	29
Description	Les API créés pour les besoins du projet devront être compatibles avec la syntaxe oData du champs \$filter en tenant compte des besoins projet pour ne pas implémenter l'ensemble des fonctions. L'implémentation de ces opérateurs doit utiliser la librairie partagée : https://gitlab.altair.recouv/sded/coeur-de-metier-recouvrement/controle-et-recouvrement-amiable-force-parametres/raf/raf-renove/transverse/raf-lib-filtrage-odata Les champs non exposés dans les ressources ne doivent pas être disponible via odata (voir anotation hidden) Les filtres odata doivent être documentés dans les swagger.

1.1.4.2.1.2.3 API Backend - Certificats et exposition Archimed

Étiquette du champ	Valeur du champ
ID de l'exigence	872
Nom	API Backend - Certificats et exposition Archimed
Etat exigence	24
Description	Les applications clientes d'un service ou exposant un service sur Archimed doivent comporter des éléments tels que : - Les magasins de certificats (Keystore/Truststore) qui devront contenir les certificats des autorités racines (AC) de l'acoss et qui sont protégés par un mot de passe correspondant au clientSecret de l'application - La/Les paire(s) de clés publiques/privées nécessaires pour s'authentifier lors des appels à la gateway Note: Pour ce faire le dépôt de référence contient 2 scripts appelés depuis le script deploy.sh et qui sont chargés de créer le magasin de clés et de récupérer la/les paires de clés. Note: En développement les clientId et secret sont obtenus lors du premier déploiement du bundle archimed

1.1.4.2.1.2.4 API Backend - Consommation de services externes

Étiquette du champ	Valeur du champ
ID de l'exigence	873
Nom	API Backend - Consommation de services externes
Etat exigence	27
Description	Le code de l' API permettant d'interroger un service externe ne doit pas être présentes dans les sources. Il doit être généré à partir du fichier descriptif de ce service (swagger, wsdl) ou être contenu dans des librairies importées Note: Utiliser openAPI pour les swaggers

1.1.4.2.1.2.5 API Backend - Localisation des contracts de service exposés

Étiquette du champ	Valeur du champ
ID de l'exigence	874
Nom	API Backend - Localisation des contracts de service exposés
Etat exigence	27
Description	Chaque composant applicatif de type API contient dans ses sources les fichiers descriptifs (ex: swagger) des API qu'il expose. Ces derniers sont déposés sur nexus comme un artefact complémentaire de l'application. Le nom du fichier doit comprendre le terme "openapi" pour être détecté par gitlab en tant que contrat.

1.1.4.2.1.2.6 API Backend - Récupération des contrats de services consommés

Étiquette du champ	Valeur du champ
ID de l'exigence	875
Nom	API Backend - Récupération des contrats de services consommés
Etat exigence	26
Description	Lorsqu'un composant applicatif consomme une API backend, Alors il ne doit pas posséder dans ses sources les fichiers descriptif du service (ex: swagger). Il doit les récupérer directement sur Nexus.

1.1.4.2.1.2.7 API Backend - Granularité des API

Étiquette du champ	Valeur du champ
ID de l'exigence	1797
Nom	API Backend - Granularité des API
Etat exigence	13
Description	• Une ressource doit être en priorité de granularité fine. • Elle ne doit contenir de sous-ressources sauf exception validés en phase de conception. • Les sous ressources et onetoMany ne peuvent contenir que des id de référence vers ces dernières.

- Les ressources ne doivent pas contenir des champs représentant l'id du parent. Les ressources comprenant des sous-ressources en grand nombre ne doivent pas avoir une liste de ces objets dans un champ, c'est-à-dire qu'elles doivent être requêtées séparément.
- Une ressource avec des sous-ressources limitées peut comprendre un champ de type liste de sous-ressources, mais ces objets doivent être simplifiés et ne contenir que l'id.

1.1.4.2.1.2.8 API Backend - Pagination des listes

Étiquette du champ	Valeur du champ
ID de l'exigence	1801
Nom	API Backend - Pagination des listes
Etat exigence	14
Description	Les APIs retournant des listes doivent gérer une pagination avec un nombre d'enregistrement par défaut pour empêcher le requêtage sans limite des tables en l'absence de paramètre dans la requête. Une valeur maximale de nombre d'éléments doit aussi être mise en place. Conformément aux exemples fullstack, il faut utiliser le champ "x-spring-paginated=true" dans les contrats afin de générer des contrôles Spring Boot intégrant la pagination et retournant des objets Pageable.

1.1.4.2.1.2.9 API Backend - OData requêtes complexes en deux appels

Étiquette du champ	Valeur du champ
ID de l'exigence	1902
Nom	API Backend - OData requêtes complexes en deux appels
Etat exigence	14
Description	Les requêtes OData sont parfois trop longues pour les chaînes de requête param. Le pattern à appliquer dans ce cas est la mise en place d'un premier appel POST sur une pseudo ressource /filtre pour stocker la requête. Le corps de la requête contient la chaîne de filtre telle qu'elle aurait été envoyée dans le paramètre \$filter. Le filtre est ainsi stocké côté serveur et un identifiant est retourné au client dans un objet json. <pre>{ "idFiltreRecherche": "string" }</pre> Le second appel est fait sur la ressource souhaitée comme cela aurait été le cas pour une requête classique avec filtre, mais au lieu de remplir le champ \$filter, on passe l'id précédemment reçu dans le paramètre : idFiltreRecherche. L'utilisation du pattern en deux appels doit être appliquée uniquement lorsque c'est nécessaire. Pour cela, il faut mesurer la longueur de la chaîne url + filtre qui doit être < 2000 caractères.

1.1.4.2.1.2.10 API Backend - Consommation de services externes - cache et throttling

Étiquette du champ	Valeur du champ
ID de l'exigence	2416
Nom	API Backend - Consommation de services externes - cache et throttling
Etat exigence	8
Description	Les services utilisant des api doivent implémenter un système de cache adapté à l'utilisation qui est faite des données et à leur validité afin de minimiser l'impact sur les service externes et améliorer les performance globales du client. Certaines api peuvent avoir un mécanisme de throttling lorsque c'est le cas il faut gérer l'erreur http 429 en conséquence avec potentiellement la mise en place d'un retry ou l'adaptation du nombre d'appel par secondes dans le cas de batch.

1.1.4.2.1.2.11 API Backend - Ressources historisées

Étiquette du champ	Valeur du champ
ID de l'exigence	3209
Nom	API Backend - Ressources historisées
Etat exigence	5
Description	Etat d'une ressource L'état d'une ressource peut être représenté de la manière suivante: Etat simple = Attribut simple de la ressource: 1 énuméré (<=> 1 colonnes en BDD) => cf ENF 1638 Ne permet pas l'historisation spécifique de l'état uniquement Etat complexe = Ressource EtatRessource dédié: 1 Ressource dédié pour représenter l'état d'une ressource L'id de l'état devient un uuid normal (n'est pas une constante) Le nom est obligatoire et doit être un énuméré (une constante utilisable dans le code pour cibler l'état d'une ressource) Permet de stocker plusieurs informations liées/représentation l'état Permet l'historisation de l'état d'une ressource sans historiser toute la ressource Api REST Toutes les ressources exposées par Api REST doivent respecter la norme restfull décrite par openapi 3.0. Dans ce qui suit nous allons utiliser comme exemple une ressource, sans historisation fonctionnelle, qui se nomme "RessourceX" qui est lié à: - un état de type complexe: - sous ressource - notion d'état courant/applicable) - avec historisation fonctionnelle - nommé "EtatRessourceX"

- une sous ressource lié
 - relation "1..n"
 - sans notion de courant/applicable
 - nommé "SousR"

Les règles de nommages des endpoint suivantes doivent être respecté:

nommage snake_case

GET Simple /ressource_xs/:id

GET All /ressource_xs

GET Filtrage complexe /ressource_xs?filtre="..." (odata) NON

Réponse partielle Aucune sous ressource n'est remonté NON Sur demande on peut faire remonter la liste des ids de sous ressources (ou une sus ressource entière, attention aux listes).

Pagination

Nom paramètre etat_courant

Exemples de endpoint (en suivant l'exemple cité):

/ressource_xs?filtre="..." Renvoie la liste des ressources suivant le filtre en paramètre
NON

/ressource_xs/:id Renvoie la ressource sans ses sous ressource (sans l'état)
OUI

/ressource_xs/:id/sous_rs Renvoie la liste sous ressource SousR lié à la ressource
NON

/ressource_xs/:id/etats Renvoie la liste de tous l'historique de l'état de la ressource
OUI

/ressource_xs/:id/etats?filtre="..." Renvoie la liste des états de la ressources suivant le filtre en paramètre
NON

Les "shortcut" suivant sont possible/conseillé dans les endpoint:

=> Voir pour enrichir la lib odata actuelle avec des mots clé pour ne rien faire en spécifique dans un produit final au niveau endpoint et juste redéfinir la fonction qui fait le filtre (si nécessaire), dans la limite de la norme OData (https://docs.oasis-open.org/odata/odata/v4.0/errata03/os/complete/part2-url-conventions/odata-v4.0-errata03-os-part2-url-conventions-complete.html#_Toc453752358)

=> Shortcut possible si on n'arrive pas a construire un filtre odata pour répondre au besoin

1.1.4.2.1.3 Batch

Étiquette du champ	Valeur du champ
ID de l'exigence	877
Nom	Batch
Etat exigence	10
Description	

1.1.4.2.1.3.1 Batch - Framework

Étiquette du champ	Valeur du champ
ID de l'exigence	878
Nom	Batch - Framework
Etat exigence	29
Description	Le framework Spring Batch est utilisé pour implémenter les batch. Il fournit une implémentation mature, robuste et maintenable des couches, composants et services habituellement utilisés dans les systèmes mainframes pour créer des applications batch.

1.1.4.2.1.3.2 Batch - Planification batch multi applicatifs

Étiquette du champ	Valeur du champ
ID de l'exigence	879
Nom	Batch - Planification batch multi applicatifs
Etat exigence	22
Description	La planification des batch complexes multi applicatifs doit passer par l'outil Automator qui est la solution retenue par la DSI pour l'ordonnancement et la planification des tâches informatiques. Il permet d'enchaîner des travaux selon un schéma de production à partir de critères variés : état d'une tâche précédente, événement particulier, action d'un utilisateur, calendrier et horaire. Il permet aussi d'enchaîner les travaux entre différentes applications. Note: Automator étant encore en cours de mise en place il n'existe pas de norme ou documentation pour l'intégrer.

1.1.4.2.1.3.3 Batch/SNV2 - Planification batch régionaux

Étiquette du champ	Valeur du champ
ID de l'exigence	880
Nom	Batch/SNV2 - Planification batch régionaux
Etat exigence	22
Description	Saturne est l'application d'administration et d'ordonnancement nationale de la branche. Il doit être utilisé pour les batchs régionaux utilisant la base SNV2.

1.1.4.2.1.3.4 Batch/SNV2 - Création commande

Étiquette du champ	Valeur du champ
ID de l'exigence	881
Nom	Batch/SNV2 - Création commande
Etat exigence	22
Description	Lors de la création ou mise à jour d'une commande (création ou modification de paramètres PRA1), le développeur doit suivre la procédure de la documentation GLD suivante : Référence : Documentation GLD (chapitre 6.11. MISE A JOUR DES COMMANDES)

1.1.4.2.1.3.5 Batch/PFS - Planification batch nationaux via Automator

Étiquette du champ	Valeur du champ
ID de l'exigence	882
Nom	Batch/PFS - Planification batch nationaux via Automator
Etat exigence	27
Description	La planification de batchs dans le cadre des applications nationales doit passer par automator. Dans le cas des applications conteneurisées et déployées sur la PFS passe par une déclaration sous forme de configmap spécifique (contenant une déclaration de type job). La fréquence et les paramètres du job devront être fournis dans le DEX afin d'être déployé dans l'outil Automator par les équipes d'exploitation. Afin de simplifier le travail en développement il est possible de construire un chart helm utilisant la valeur de l'environnement pour déployer en dev seulement un cronjob et utiliser une configmap sur les autres environnements. ----- Complément : http://confluence.urssaf.recouv/display/ENFAPORT/ENFp++Ordonnancer+uniquement+via+AUTOMATOR

1.1.4.2.1.3.6 Batch/Hawai - Planification

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	883
Nom	Batch/Hawai - Planification
Etat exigence	22
Description	La planification sur Hwai passe par des crontab afin d'être administrable via webmin

1.1.4.2.1.3.7 Batch backend/PFS - Déclaration dans le charthelm

Étiquette du champ	Valeur du champ
ID de l'exigence	1420
Nom	Batch backend/PFS - Déclaration dans le charthelm
Etat exigence	19
Description	Un traitement batch qui possède des paramètres de lancement doit être déclaré dans une ressource de type "configmap". Ce configmap doit posséder le paramètre "data.job" qui a pour valeur la ressource du job. Chaque paramètre est identifié par une chaîne de caractères doit avoir la syntaxe suivante : "@<nom du paramètre>@"

1.1.4.2.1.3.8 Batch backend/PFS - Configuration environnement dev/autre

Étiquette du champ	Valeur du champ
ID de l'exigence	1421
Nom	Batch backend/PFS - Configuration environnement dev/autre
Etat exigence	19
Description	En environnement de dev : utiliser une ressource de de type cronjob. Il ne doit pas être déployé sur l'environnement de prod. En environnement de prod : utiliser une ressource de type job

1.1.4.2.2 Frontend

Étiquette du champ	Valeur du champ
ID de l'exigence	884
Nom	Frontend
Etat exigence	7
Description	Ceci est un test d'injection d'une exigence

1.1.4.2.2.1 Frontend - Mise en place de BIPE

Étiquette du champ	Valeur du champ
ID de l'exigence	2405
Nom	Frontend - Mise en place de BIPE

Etat exigence	7
Description	
Pour chaque ecran d'une ihm développé il doit être implémenté un descripteur BIPE minimal pour y accéder.	

1.1.4.2.3 Base de données

Étiquette du champ	Valeur du champ
ID de l'exigence	885
Nom	Base de données
Etat exigence	13
Description	

1.1.4.2.3.1 Base de données/SNV2 - Evolution de schéma

Étiquette du champ	Valeur du champ
ID de l'exigence	886
Nom	Base de données/SNV2 - Evolution de schéma
Etat exigence	13
Description	
Toute création, modification ou suppression de la structure de la base de données SNV2 devra se faire à travers l'outil Liquibase selon les préconisations de la documentation sx5 fournie en référence. Note: Les fichiers liquibase seront au format UTF8 seront stockés dans le dossier /sx/appv2/bdd/raf-parcours du dépôt cntx.	

1.1.4.2.3.2 Base de données/SNV2 - Nature de la base de données - Migration

Étiquette du champ	Valeur du champ
ID de l'exigence	887
Nom	Base de données/SNV2 - Nature de la base de données - Migration
Etat exigence	8
Description	
La solution doit être compatible avec les SGBD : Oracle et PostgreSQL pour pouvoir suivre la migration en cours vers PostgreSQL selon les recommandations du document ACROSS-DAT-SX-PG	

1.1.4.2.3.3 Base de données/SNV2 - Utilisation de fonctions standards - Migration

Étiquette du champ	Valeur du champ
ID de l'exigence	888
Nom	Base de données/SNV2 - Utilisation de fonctions standards - Migration
Etat exigence	8
Description	
Les développements ne devront pas utiliser de fonctions spécifiques aux implémentations des éditeurs afin de pouvoir être réutilisés sans contraintes dans d'autres implémentations.	

1.1.4.2.3.4 Base de données/SNV2 - Bascule des SGDB - Migration

Étiquette du champ	Valeur du champ
ID de l'exigence	889
Nom	Base de données/SNV2 - Bascule des SGDB - Migration
Etat exigence	8
Description	La bascule entre les SGBD (Oracle/Postgres) doit pouvoir se faire à l'exécution sur la base de la variable système "SGBD_ACTIF" ou de la réponse d'un point d'api.

1.1.4.2.3.5 Base de donnée/PFS - Evolution de schéma

Étiquette du champ	Valeur du champ
ID de l'exigence	890
Nom	Base de donnée/PFS - Evolution de schéma
Etat exigence	13
Description	Toute création, modification ou suppression de la structure de la base de données PFS devra se faire à travers l'outil Liquibase selon les préconisations de la documentation de référence. L'exécution de cette modification de schéma doit être exécutée par un init container spécifique liquibase avant le démarrage du pod applicatif.

1.1.4.2.3.6 Base de données - Procédure de restauration

Étiquette du champ	Valeur du champ
ID de l'exigence	891
Nom	Base de données - Procédure de restauration
Etat exigence	8
Description	La base de données peut être restaurée en cas d'erreur lors d'un processus batch existant ou lors d'une demande de l'Agent Comptable. La solution doit considérer ces possibles restaurations et le cas échéant retrouver un état fonctionnel cohérent.

1.1.4.2.3.7 Base de données - Configuration de la disponibilité

Étiquette du champ	Valeur du champ
ID de l'exigence	892
Nom	Base de données - Configuration de la disponibilité
Etat exigence	8
Description	Le cluster PostgreSQL doit être configuré avec les options de haute disponibilité

1.1.4.2.3.8 Base de données - Procédure de backup

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	893
Nom	Base de données - Procédure de backup
Etat exigence	8
Description	Une procédure et un script Shell pour permettre de récupérer des backups devront être fournis à la production

1.1.4.2.3.9 Base de donnée/PFS - Ouverture du port

Étiquette du champ	Valeur du champ
ID de l'exigence	1802
Nom	Base de donnée/PFS - Ouverture du port
Etat exigence	10
Description	Les bases donnée déployées en PFS doivent pouvoir être accédées via DBBeaver. Pour cela le chart helm doit inclure la création d'un loadbalancer optionnel. Ce load balancer sera créé si la valeur suivante est à true : expositionBDD: true

1.1.4.2.3.10 Base de donnée/PFS - PostgREST

Étiquette du champ	Valeur du champ
ID de l'exigence	1804
Nom	Base de donnée/PFS - PostgREST
Etat exigence	10
Description	Les bases donnée déployées en PFS doivent pouvoir être accédées en rest en déployant un composant PostgREST dans le chart. Ce dernier doit aussi être assujéti a une variable pour l'activer.

1.1.4.2.3.11 Base de données - Procédure de restauration_Copie_1

Étiquette du champ	Valeur du champ
ID de l'exigence	3212
Nom	Base de données - Procédure de restauration_Copie_1
Etat exigence	3
Description	La base de données peut être restaurée en cas d'erreur lors d'un processus batch existant ou lors d'une demande de l'Agent Comptable. La solution doit considérer ces possibles restaurations et le cas échéant retrouver un état fonctionnel cohérent.

1.1.4.2.4 Bus de message

Étiquette du champ	Valeur du champ
ID de l'exigence	894

Nom	Bus de message
Etat exigence	27
Description	

1.1.4.2.4.1 Bus de message - Utilisation Kafka

Étiquette du champ	Valeur du champ
ID de l'exigence	895
Nom	Bus de message - Utilisation Kafka
Etat exigence	40
Description	
	Le bus de message utilisé par l'application doit être la solution Kafka

1.1.4.2.4.2 Bus de message - Se connecter à Kafka de manière sécurisée

Étiquette du champ	Valeur du champ
ID de l'exigence	1512
Nom	Bus de message - Se connecter à Kafka de manière sécurisée
Etat exigence	54
Description	
	Les applications doivent accéder à Kafka de manière sécurisée et c'est recommandé d'utiliser la librairie Strimzi qui va gérer pour vous toute la mécanique OAuth2. https://readthedocs.cnp.recouv/docs/fullstack/fr/latest/fullstack/fullstack-rest-backend/plusloin/kafka_acl.html#se-connecter-a-kafka-de-maniere-securisee

1.1.4.2.4.3 Bus de message - Définition des topics

Étiquette du champ	Valeur du champ
ID de l'exigence	896
Nom	Bus de message - Définition des topics
Etat exigence	48
Description	
	La norme Kafka demande de respecter la nomenclature suivante pour les topics : "couloir.domaine.nomDuTopicEnCamelCase". • couloir permet de mettre plusieurs versions d'un même topic sur un même environnement. Cela permet au développeur travaillant sur plusieurs branches de pouvoir travailler en parallèle. En production le couloir n'est pas présent. Ex : dev1, dev2... • domaine permet d'indiquer le domaine fonctionnel sur lequel porte le topic • nomDuTopicEnCamelCase doit être le plus parlant possible, notamment car d'autres MOE pourraient être consommatrice de ce topic

1.1.4.2.4.4 Bus de message - Variabilisation des topics

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	897
Nom	Bus de message - Variabilisation des topics
Etat exigence	29
Description	Les noms de TOPIC Kafka doivent être variabilisés : • Avec des variables Castor en SNV2 • Avec des tokens avec HAWAI • Avec des variables helm pour la PFS ({{Values.XXX}}) Note: la nomenclature des topics est spécifié dans une exigence spécifique

1.1.4.2.4.5 Bus de message - Sécuriser l'accès aux topics en utilisant les ACL

Étiquette du champ	Valeur du champ
ID de l'exigence	1513
Nom	Bus de message - Sécuriser l'accès aux topics en utilisant les ACL
Etat exigence	64
Description	Pour sécuriser l'accès aux topics de l'application, il faut définir la liste des composants qui seront autorisés à y accéder ainsi que les opérations qui pourront être réalisées. Il faudra aussi autoriser voir application à utiliser ses propres topics. https://readthedocs.cnp.recouv/docs/fullstack/fr/latest/fullstack/fullstack-rest-backend/plusloin/kafka_acl.html#securiser-l-acces-a-vos-topics

1.1.4.2.4.6 Bus de messages - Format des messages

Étiquette du champ	Valeur du champ
ID de l'exigence	898
Nom	Bus de messages - Format des messages
Etat exigence	28
Description	Les messages doivent être au format binaire Avro (http://avro.tarantool.org/). Le fichier AVRO doit être présent dans les sources. Les règles suivantes doivent être suivies:: • Si un paramètre est associé à une valeur parmi une liste de valeurs possibles, Alors il faut utiliser un énuméré • La logique de nommage des paramètres doit être cohérente • L'attribut "doc" doit contenir: • Une description sémantique et les contraintes associées aux différents paramètres • L'utilité du message

1.1.4.2.4.7 Bus de messages - Structure d'un messages fonctionnel

Étiquette du champ	Valeur du champ
ID de l'exigence	899
Nom	Bus de messages - Structure d'un messages fonctionnel
Etat exigence	30
Description	
Pour les messages fonctionnels, les champs nécessaires à la traçabilité (correlationId, recouvid) et au suivi doivent être décrits dans les spécifications détaillées. Leur évolution doit rester possible et doit être facilitée par l'utilisation d'héritage. Ceci pourrait arriver dans le futur en cas de mise en place de nouveaux outils transverse dans le SI et de création de normes associées. Note: Le contenu spécifique des messages sera contenu dans la partie body et déterminé selon le besoin Le message doit avoir la structure suivante: • Présence d'un bloc "header" contenant : ○ Attribut "id du message" ○ Attribut "correlationId" unique à générer à chaque message ○ Attribut "recouvid" peut reprendre un correlationId d'un message ou d'une requête http si le message est publié dans le cadre du traitement de ces dernier, sinon positionné à la même valeur que correlationID ○ Attribut "Origine (TP ou Batch)" ○ Attribut "Code URSSAF" ○ Attribut "Date du commit" ○ Attribut "Date du messageCode libellé du messageClés (fonctionnel ou technique de l'objet ciblé)" ○ Attribut "Qualité du message" • Présence d'un bloc "body" contenant: ○ les données métier	

1.1.4.2.4.8 Bus de messages - Désérialisation des messages

Étiquette du champ	Valeur du champ
ID de l'exigence	900
Nom	Bus de messages - Désérialisation des messages
Etat exigence	28
Description	
La méthode de deserialisation des messages kafka doit être surchargée pour gérer les erreurs de la manière suivante : • Appeler la méthode deserialize de la classe mère KafkaAvroDeserializer. • En cas d'erreur (exception), générer les logs suivants : • ERROR : Type et message de l'exception • INFO : Détails : topic, contenu en mode String et en mode binaire (tableau d'octets) • Retourner un enregistrement (GenericData.Record) contenant un seul champ ERROR avec le type et message de l'exception.	

1.1.4.2.4.9 Bus de messages - Versioning des messages

Étiquette du champ	Valeur du champ
ID de l'exigence	901
Nom	Bus de messages - Versioning des messages
Etat exigence	27

Description
Chaque message doit contenir un champ version au même niveau que les champs « doc », « name », « namespace », « type » La version est une chaîne de caractère comportant 3 digits : "Majeur.mineur.patch" (ex:2.0.1)

1.1.4.2.4.10 Bus de messages - Stratégie de développement

Étiquette du champ	Valeur du champ
ID de l'exigence	902
Nom	Bus de messages - Stratégie de développement
Etat exigence	26
Description	
Les développements Kafka doivent se faire en Contract First : La définition de la structure du message est donc à réaliser avant l'implémentation dans le code.	

1.1.4.2.4.11 Bus de message/Inter-domaine - Ajout fournisseur

Étiquette du champ	Valeur du champ
ID de l'exigence	903
Nom	Bus de message/Inter-domaine - Ajout fournisseur
Etat exigence	16
Description	
Dans le cas de topic inter-domaines, l'ajout d'un nouveau fournisseur doit être validé au préalable par les architectes	

1.1.4.2.4.12 Bus de message/Inter-domaine - Abonnement à un topic sensible

Étiquette du champ	Valeur du champ
ID de l'exigence	904
Nom	Bus de message/Inter-domaine - Abonnement à un topic sensible
Etat exigence	16
Description	
Dans le cas de topic inter-domaines, si les données du topic sont considérées comme sensible, l'abonnement d'un consommateur doit être validé par la tribu PROTEINE	

1.1.4.2.4.13 Bus de messages/Inter-domaine - Vérification de grammaire

Étiquette du champ	Valeur du champ
ID de l'exigence	905
Nom	Bus de messages/Inter-domaine - Vérification de grammaire
Etat exigence	23
Description	
Dans le cas de topic inter-domaines, l'utilisation schema registry (avro) ou cloudEvents. La vérification de la grammaire est obligatoire	

1.1.4.2.4.14 Bus de message/Inter-domaine - Accès topic sécurisé

Étiquette du champ	Valeur du champ
ID de l'exigence	906
Nom	Bus de message/Inter-domaine - Accès topic sécurisé
Etat exigence	16
Description	Dans le cas de topic inter-domaines, seuls les composants ayant un besoin impérieux, validé par la tribu d'architectes (PROTEINE), peuvent accéder en fournisseur ou consommateur à ce topic sécurisé Note: Ils peuvent être d'un même domaine ou de domaines différents

1.1.4.2.4.15 Bus de message/Intra-domaine - Accès topic

Étiquette du champ	Valeur du champ
ID de l'exigence	907
Nom	Bus de message/Intra-domaine - Accès topic
Etat exigence	16
Description	Dans le cas de topic intra-domaines, l'accès au topic par un nouveau composant du domaine est sous responsabilité de l'architecte du domaine

1.1.4.2.4.16 Bus de message/Intra-Domaine - Accès topic sensible

Étiquette du champ	Valeur du champ
ID de l'exigence	908
Nom	Bus de message/Intra-Domaine - Accès topic sensible
Etat exigence	20
Description	Dans le cas de topics Intra-Domaine, si les données sont considérées comme sensibles, alors l'abonnement d'un consommateur doit être validé par l'architecte Note: Seuls les composants ayant un besoin impérieux, validé par la tribu d'architectes (PROTEINE), peuvent accéder en fournisseur ou consommateur à ce topic sécurisé

1.1.4.2.4.17 Bus de messages - Volumétrie

Étiquette du champ	Valeur du champ
ID de l'exigence	909
Nom	Bus de messages - Volumétrie
Etat exigence	24
Description	Les messages transmis via kafka à destination de l'ensemble du SI doivent rester légers . Note: C'est à dire qu'ils doivent se limiter à des ID ou quelques champs comme un code événement. Il faut éviter les grappes d'objets.

1.1.4.2.4.18 Bus de messages - Publication de type Ressource

Étiquette du champ	Valeur du champ
ID de l'exigence	2415
Nom	Bus de messages - Publication de type Ressource
Etat exigence	6
Description	Lorsque demandée par l'architecture, la publication d'évènement ressource doit suivre les principes suivants : L'objectif est d'informer d'un changement de la Ressource (Create, Update, Delete). Ce type de publication offre un spectre de couverture de besoins très large car il ne présume pas de l'usage fonctionnel qui sera fait par le consommateur. Toutefois, du point de vue des consommateurs, une action de filtrage sera probablement nécessaire. 2 Règles de déclenchement - La publication est pilotée à travers les opérations Create, Update, Delete - Publier tous les messages liés aux opérations Create, Update, Delete sur la ressource. Il ne faut pas présumer des consommateurs pour filtrer les messages envoyés. - S'assurer de la réalité d'une mise à jour (écarter le remplacement de la valeur d'une propriété par une valeur identique) pour minimiser la publication de message ressource. 2.1.1 Message de publication « Ressource » En plus de l'identification de la ressource, pour faciliter le traitement du message consommé, on ajoutera : - Les informations sur la nature de l'évènement qui a mené au message : Opération réalisée (Create, Update, Delete) - Les informations de l'impact sur la ressource dans le cas où l'impact est une mise à jour : Liste des attributs de la ressource ayant été impactés sans leur valeur. - Dans le cas de la publication « Ressource » en tant que message « Response » de publication « Request / Response », l'identifiant du message « Request » qui a provoqué l'action.

2.1.1.1.1.1 Bus de messages - Header technique

Étiquette du champ	Valeur du champ
ID de l'exigence	2529
Nom	Bus de messages - Header technique
Etat exigence	5
Description	Tous les messages transmis via kafka (qu'ils soient interne au domaine applicatif ou à destination de l'ensemble du SI) doivent inclure les données suivantes dans le header Kafka: - correlationId: UUID générée à la création du message kafka (par le producteur du message) - dateMessage: date générée à la création du message kafka (par le producteur du message), au format ISO 8601 - typePublication: enum string qui peut prendre les valeurs suivante: "Ressource": dans le cas d'un message standard sur une ressource manipulé par l'applicatif "Fonctionnel": Dans le cas d'un message indiquant un comportement spécifique fonctionnel non corrélé à une ressource

- "Request", "Response": dans le cas d'un échange similaire à une requête de type API HTTP

La norme applicative générale décrite plus haut reste applicable et contient des headers complémentaire à ajouter, par ex: ipBackend, nomBackend (=code Mega), recouv.id, etc ...

2.1.1.1.2 Transverse

Étiquette du champ	Valeur du champ
ID de l'exigence	910
Nom	Transverse
Etat exigence	10
Description	

2.1.1.1.2.1 Observabilité

Étiquette du champ	Valeur du champ
ID de l'exigence	911
Nom	Observabilité
Etat exigence	31
Description	
Ceci est un test d'injection d'une exigence	

2.1.1.1.2.1.1 Log - Framework de gestion des logs

Étiquette du champ	Valeur du champ
ID de l'exigence	912
Nom	Log - Framework de gestion des logs
Etat exigence	33
Description	
Le framework d'émission de log doit être logback	

2.1.1.1.2.1.2 Log/PFS - Gestion des logs

Étiquette du champ	Valeur du champ
ID de l'exigence	913
Nom	Log/PFS - Gestion des logs
Etat exigence	13
Description	
Sur l'environnement d'exécution PFS, les fonctionnalités de gestion des logs fournies par kubernetes sont utilisées.	

2.1.1.1.2.1.3 Log/SNV2 - Gestion des logs

Étiquette du champ	Valeur du champ
ID de l'exigence	914

Nom	Log/SNV2 - Gestion des logs
Etat exigence	11
Description	Sur l'environnement d'exécution SX, les logs doivent être stockés dans un fichier présent sur un point de montage.

2.1.1.1.2.1.4 Log - Niveaux de sévérité

Étiquette du champ	Valeur du champ
ID de l'exigence	915
Nom	Log - Niveaux de sévérité
Etat exigence	16
Description	Les logs émis doivent pouvoir être distingués selon leur niveau : • TRACE: Ce niveau est à destination des Développeurs et uniquement eux. Il a pour vocation de se substituer à un mode debug impossible sur un serveur en production et d'en retranscrire chaque changement de valeur • DEBUG: Ce niveau est à destination des Admin-sys et éventuellement des Développeurs. Si le niveau TRACE correspond à un debugger activé en mode pas-à-pas, le niveau DEBUG se contenterait d'afficher uniquement les valeurs aux points d'arrêts. • INFO: Ce niveau est à destination de Tous. Il a pour objectif de tracer les appels aux traitements métiers. • WARN: Ce niveau est à destination de Tous. Il a pour objectif d'enregistrer l'occurrence d'une erreur tout en sachant que la stabilité de l'application, la cohérence et l'intégrité de ses données ne sont pas atteintes, • ERROR/FATAL/CRITIACL: Ces niveaux sont à destination de Tous. Ils ont pour objectif de tracer qu'une erreur grave s'est produite et que l'application est potentiellement en état instable. • FATAL : erreurs techniques empêchant le lancement de la brique • CRITICAL : Service dégradé • ERROR : erreurs fonctionnelle sérieuses empêchant le traitement d'un message (abandon du traitement)

2.1.1.1.2.1.5 Log - Langue des messages

Étiquette du champ	Valeur du champ
ID de l'exigence	916
Nom	Log - Langue des messages
Etat exigence	45
Description	La langue des messages de log sont en français.

2.1.1.1.2.1.6 Log - Message techniques sans informations personnelles

Étiquette du champ	Valeur du champ
ID de l'exigence	917
Nom	Log - Message techniques sans informations personnelles
Etat exigence	27
Description	Les logs techniques de la solution ne doivent pas contenir d'informations personnelles tel que nom, prénom, nir, etc ...

Justification: Ces informations peuvent être enregistrées sur les canaux de log de traçabilité : empreinte.

2.1.1.1.2.1.7 Log - Contextualisation du message

Étiquette du champ	Valeur du champ
ID de l'exigence	918
Nom	Log - Contextualisation du message
Etat exigence	27
Description	Les logs doivent contenir un maximum d'informations permettant de reconstituer le contexte. Lorsqu'ils sont disponibles les champs suivants sont ajouté dans le log : • Identifiant de corrélation de la transaction • Code UR • Type et UUID de l'entité manipulée • UUID de l'exception • Stack trace • Topic • Client_id • Checksum du token Lorsque c'est applicable, ces informations doivent être séparés du message et stockées dans des champs dédiés.

2.1.1.1.2.1.8 Log - Ségrégation des messages

Étiquette du champ	Valeur du champ
ID de l'exigence	919
Nom	Log - Ségrégation des messages
Etat exigence	34
Description	Dans une même fonction on ne doit pas retrouver 2 fois le même message de log sans moyen de les distinguer.

2.1.1.1.2.1.9 Configuration - Niveau de sévérité des log

Étiquette du champ	Valeur du champ
ID de l'exigence	920
Nom	Configuration - Niveau de sévérité des log
Etat exigence	9
Description	Le niveau de log doit pouvoir être configuré indépendamment au niveau de chaque composant applicatif.

2.1.1.1.2.1.10 Configuration - Chargement à chaud du niveau de log

Étiquette du champ	Valeur du champ
ID de l'exigence	921

Nom	Configuration - Chargement à chaud du niveau de log
Etat exigence	9
Description	Le niveau de log d'un composant applicatif doit pouvoir être modifié lors de son exécution. Justification: Ce chargement à chaud permet d'analyser rapidement des workflow ou des anomalies

2.1.1.1.2.1.11 Supervision - Publication des métriques supervisées

Étiquette du champ	Valeur du champ
ID de l'exigence	923
Nom	Supervision - Publication des métriques supervisées
Etat exigence	13
Description	Les informations de base doivent être remontées dans les logs en niveau INFO. Les métriques spécifiques de l'application devront être mise à disposition via micrometer et décrites dans le DEX.

2.1.1.1.2.1.12 Supervision - Métriques suivies par la supervision

Étiquette du champ	Valeur du champ
ID de l'exigence	924
Nom	Supervision - Métriques suivies par la supervision
Etat exigence	13
Description	Les indicateurs supervisées sont à minima : • Volume de messages • Temps de traitement d'un message • Volume de données d'un batch • Durée d'exécution d'un batch • Durée d'exécution d'une API • Temps d'appel à un service tiers

2.1.1.1.2.1.13 Supervision/PFS - Outillage de supervision

Étiquette du champ	Valeur du champ
ID de l'exigence	925
Nom	Supervision/PFS - Outillage de supervision
Etat exigence	13
Description	Sur la PFS les outils d'exposition des métriques à utiliser sont: • Prometheus qui va venir récupérer ("scrappe") les métriques exposées à intervalle régulier • Grafana qui va permettre d'exposer un tableau de bord constitué de graphique réalisés à l'aide de ces métriques

2.1.1.1.2.1.14 Log - Log minimum

Étiquette du champ	Valeur du champ
ID de l'exigence	1459
Nom	Log - Log minimum
Etat exigence	92
Description	
Les éléments suivants doivent être logués par les applicatifs : • Appel d'une méthode au niveau d'un contrôleur (point d'entrée api, ou traitement de message) - INFO • Passage dans une méthode métier complexe ou dans une condition structurante - INFO • Passage dans une méthode - DEBUG • Emission d'une exception - ERROR	

2.1.1.1.2.1.15 Supervision/PFS - Healthcheck

Étiquette du champ	Valeur du champ
ID de l'exigence	2414
Nom	Supervision/PFS - Healthcheck
Etat exigence	71
Description	
Chaque agent doit fournir une API de type "healthcheck" exposé via Ingress. Chaque service healthcheck doit être accessible sur la gateway Archimède via le path "/actuator/health" (à partir du lot ArchiMed 21.11) Pour chaque API Backend déployée, un test de vie doit être implémenté en utilisant la couche de supervision intégrée à Spring : Spring Boot Actuator. Par défaut, le service Actuator confirme seulement le bon fonctionnement de l'application java. Cependant, pour s'assurer que l'application est pleinement fonctionnelle, des tests supplémentaires doivent être implémentés. Ceci est réalisé par le biais de tests de vie personnalisés à définir contrôlant par exemple, la connectivité aux services tiers dont l'application dépend ou des caractéristiques du cache. Ajout le 14/02/2023 (applicable dès que possible) : le résultat du test de vie doit afficher le nom et la version du micro-service testé : <pre>{ "Service" : "<nom du service>", "Version" : "<x.y.z>", ... }</pre> Chaque service healthcheck doit respecter les spécifications suivantes : • Il doit renvoyer une réponse de type application/json • Le contenu doit avoir: ○ un attribut « status » à la racine, qui a pour valeur soit « UP » (considéré comme un healthcheck OK) ou bien n'importe quelle autre valeur (considéré comme un healthcheck KO) ○ Les informations sur liquibase ○ Les métriques de la JVM ○ Les informations sur le volume d'un éventuel cache	

2.1.1.1.2.2 Qualité

Étiquette du champ	Valeur du champ
ID de l'exigence	929
Nom	Qualité
Etat exigence	11
Description	

2.1.1.1.2.2.1 Qualité - Outillages

Étiquette du champ	Valeur du champ
ID de l'exigence	930
Nom	Qualité - Outillages
Etat exigence	32
Description	Les outils de contrôle de la qualité des sources sont sonarqube checkstyle lint (pour les application front end) Chaque composant applicatif contient dans ses sources les fichiers paramétrage des outils de qualités: le fichier de configuration sonar le fichier de configuration checkstyle le fichier de configuration lint (pour les applications frontend)

2.1.1.1.2.2.2 Qualité - Langage français

Étiquette du champ	Valeur du champ
ID de l'exigence	931
Nom	Qualité - Langage français
Etat exigence	59
Description	La langue de développement est le francais. Cela inclut : • les commentaires • les noms (méthodes, classes, projets, attributs, paramètres) Afin de faciliter la compréhension des concepts de programmation et l'association avec les librairies ou frameworks les termes dédiés doivent rester en anglais. Par exemple : • get set • Service • Controller • DAO • registry • repository • Mapper

2.1.1.1.2.2.3 Qualité - Pas de code mort

Étiquette du champ	Valeur du champ
ID de l'exigence	932
Nom	Qualité - Pas de code mort
Etat exigence	8
Description	Toute partie de code doit être accessible (pas de code mort).

2.1.1.1.2.2.4 Qualité - Documentation du code

Étiquette du champ	Valeur du champ
ID de l'exigence	933
Nom	Qualité - Documentation du code
Etat exigence	40
Description	Les règles d'auto-documentation intrinsèques au langage de programmation (exemple: commentaires exploitables par javadoc pour le langage Java) doivent être mise en oeuvre

2.1.1.1.2.2.5 Qualité - Informations minimales contenues dans la description des fonctions

Étiquette du champ	Valeur du champ
ID de l'exigence	934
Nom	Qualité - Informations minimales contenues dans la description des fonctions
Etat exigence	40
Description	Les commentaires de chaque fonction ou sous-programme doit préciser ses paramètres en entrée, en sortie ou en entrée/sortie, avec pour chacun: • son nom, • sa signification, • l'unité utilisée en cas de grandeur physique, • son degré de précision, • son domaine de valeur

2.1.1.1.2.2.6 Qualité/Hawai - Encodage sources

Étiquette du champ	Valeur du champ
ID de l'exigence	935
Nom	Qualité/Hawai - Encodage sources
Etat exigence	8
Description	Les fichiers sources doivent être disponibles dans GIT au format UTF-8 pour les développements sous HAWAI

2.1.1.1.2.2.7 Qualité - Traçage des exclusions

Étiquette du champ	Valeur du champ
ID de l'exigence	1411
Nom	Qualité - Traçage des exclusions
Etat exigence	26
Description	Le fichier sonar-project.properties doit contenir des commentaires justifiant chaque élément exclus lors des analyses

2.1.1.1.2.2.8 Qualité - Checkstyle

Étiquette du champ	Valeur du champ
ID de l'exigence	2407
Nom	Qualité - Checkstyle
Etat exigence	7
Description	Les configuration violationSeverity & failOnViolation ne doivent pas être surchargées

2.1.1.1.2.2.9 Qualité - Gestion des Exception

Étiquette du champ	Valeur du champ
ID de l'exigence	3217
Nom	Qualité - Gestion des Exception
Etat exigence	4
Description	- Les exception capturée doivent toujours être loggées - Une exception capturée doit être traité - soit par mise en oeuvre d'un code dédié permettant à la fonction de terminer nominalement avec un valeur de retour cohérente - soit en propageant l'exception - Les exception Générales (Exception, Throwable) ne doivent pas être capturées ou si elle le sont dans un cas spécifique approuvé en conception elle doivent absolument logger la stacktrace - Les try catch mis en place doivent correctement libérer les ressources (fichier, socket etc ...)

2.1.1.1.2.3 Testabilité

Étiquette du champ	Valeur du champ
ID de l'exigence	936
Nom	Testabilité
Etat exigence	6
Description	

2.1.1.1.2.3.1 Testabilité - Création de bouchon

Étiquette du champ	Valeur du champ
ID de l'exigence	937

Nom	Testabilité - Création de bouchon
Etat exigence	27
Description	La création de bouchons doit passer par un projet de type Fullstack dédié. Il bénéficiera: De son propre dépôt GIT De sa propre chaine de construction Note: Le projet sera généré à partir de la définition d'APIs d'origine et des réponses statiques ou partiellement dynamiques construites dans les controllers.

2.1.1.1.2.3.2 Testabilité - Packaging des bouchons

Étiquette du champ	Valeur du champ
ID de l'exigence	938
Nom	Testabilité - Packaging des bouchons
Etat exigence	13
Description	Le projet de type bouchon devra être packagé sous forme d'image docker pour être intégré dans les environnements de test ou sur le poste développeur.

2.1.1.1.2.3.3 Testabilité - Intégration Backend postman

Étiquette du champ	Valeur du champ
ID de l'exigence	939
Nom	Testabilité - Intégration Backend postman
Etat exigence	31
Description	Des tests postman doivent être implémentés pour chaque opération de service exposé. Toutes les informations contextuelles telles que l'url doivent être contenues dans des fichiers environnements. Il doit y avoir un jeu de fichier d'environnement pour l'environnement de développement et un par environnement d'exécution. Justification: Ces tests permettent de vérifier rapidement le bon fonctionnement des API backend déployés sur l'environnement cible ou de développement et de détecter les regressions.

2.1.1.1.2.3.4 Testabilité - Nommage des Tests Unitaires

Étiquette du champ	Valeur du champ
ID de l'exigence	1903
Nom	Testabilité - Nommage des Tests Unitaires
Etat exigence	7
Description	Pour une meilleure lisibilité, les noms des méthodes de test doivent suivre le template suivant : testMethodeATesterEtat, où MethodeATester est le nom de la méthode à tester. Etat = KO, OK ou autre état spécifique. <i>Exemple : pour tester executerEnchainement(), deux méthodes de test sont à créer : testExecuterEnchainementKO() et testExecuterEnchainementOK()</i>

--

2.1.1.1.2.4 Configurabilité

Étiquette du champ	Valeur du champ
ID de l'exigence	940
Nom	Configurabilité
Etat exigence	7
Description	

2.1.1.1.2.4.1 Configuration - Gestion par environnements d'exécution

Étiquette du champ	Valeur du champ
ID de l'exigence	941
Nom	Configuration - Gestion par environnements d'exécution
Etat exigence	60
Description	Chaque composant applicatif doit contenir dans ses sources les fichiers de configuration utilisés dans ses environnements d'exécution : l'environnement de développement (développement) l'environnement cible (ex: pfs) Les fichiers de configuration spécifiques à un environnement d'exécution doivent être suffixés par l'environnement d'exécution cible. (ex: application-pfs.yaml)

2.1.1.1.2.4.2 Configuration - Variabilisation dépendant de l'environnement d'exécution

Étiquette du champ	Valeur du champ
ID de l'exigence	942
Nom	Configuration - Variabilisation dépendant de l'environnement d'exécution
Etat exigence	13
Description	L'exécutable d'un composant applicatif de type API doit être paramétrable par environnement d'exécution. - SX : Variables Castor - HAWAI : Tokens HAWAI - PFS : Variables HELM

2.1.1.1.2.4.3 Configuration - Méta informations dans entêtes pour routing archimed

Étiquette du champ	Valeur du champ
ID de l'exigence	943
Nom	Configuration - Méta informations dans entêtes pour routing archimed
Etat exigence	13
Description	Hors production, il doit être possible, par configuration, d'ajouter ou non dans l'entête des requêtes HTTP métier transitant par Archimed une propriété "recouv.vhost". Une valeur différente doit pouvoir être configurée pour chaque service métier.

Justification:

Il existe parfois à un instant T plusieurs instances d'une même application déployées sur une plateforme (par exemple pour gérer différentes versions en parallèle ou des environnements d'intégration et de démonstration). Malheureusement, ArchiMed ne peut reconnaître qu'une instance d'une application donnée par plateforme. Toutes les requêtes transitant par Archimed sont consommée par une seule instance d'application. Pour contourner ce problème, l'assemblage est capable de prendre en compte un entête HTTP « recouv.vhost » qui permet de surcharger le routage par défaut vers l'application cible. Il est ainsi possible de faire cohabiter plusieurs instances du même backend sur la même plateforme ArchiMed.

2.1.1.1.2.4.4 Configuration - Activation Prisme

Étiquette du champ	Valeur du champ
ID de l'exigence	944
Nom	Configuration - Activation Prisme
Etat exigence	17
Description	Il doit être possible de configurer l'activation ou la désactivation de prisme au niveau d'un composant applicatif

2.1.1.1.2.4.5 Configuration - Propriétés dans les entêtes HTTP

Étiquette du champ	Valeur du champ
ID de l'exigence	945
Nom	Configuration - Propriétés dans les entêtes HTTP
Etat exigence	15
Description	Il doit être possible par configuration d'ajouter des propriétés dans les headers des requêtes http.

2.1.1.1.2.5 COTS

Étiquette du champ	Valeur du champ
ID de l'exigence	946
Nom	COTS
Etat exigence	3
Description	

2.1.1.1.2.5.1 COTS - Licence Open Source non contaminante

Étiquette du champ	Valeur du champ
ID de l'exigence	947
Nom	COTS - Licence Open Source non contaminante
Etat exigence	17
Description	Le fabricant s'engage à ne pas intégrer de logiciels Open Source dont les licences sont incompatibles entre elles ou avec les licences existantes, et à ce que les licences OpenSource utilisées ne soient pas de nature à contaminer les logiciels sous licences propriétaires développés ou adaptés dans le cadre de cette prestation.

--

2.1.1.1.2.6 Sécurité

Étiquette du champ	Valeur du champ
ID de l'exigence	948
Nom	Sécurité
Etat exigence	5
Description	

2.1.1.1.2.6.1 Sécurité - Analyse des vulnérabilités durant tout le cycle de vie du produit

Étiquette du champ	Valeur du champ
ID de l'exigence	949
Nom	Sécurité - Analyse des vulnérabilités durant tout le cycle de vie du produit
Etat exigence	13
Description	La recherche de vulnérabilités dans les images doivent être effectuée de manière régulière tout au long de la vie du produit. Les nouvelles failles critiques découvertes devront être prises en compte et corrigées par la partie responsable de l'appliquatif au moment de la vérification.

2.1.1.1.2.6.2 Sécurité - Appels HTTPS

Étiquette du champ	Valeur du champ
ID de l'exigence	1423
Nom	Sécurité - Appels HTTPS
Etat exigence	21
Description	Les appels à Archimed doivent se faire en HTTPS.

2.1.1.1.2.6.3 Sécurité - Charte

Étiquette du champ	Valeur du champ
ID de l'exigence	3210
Nom	Sécurité - Charte
Etat exigence	4
Description	La sécurité des développements doit être assurée conformément à l'état de l'art et aux règles des normes de sécurité fournies. Ci-dessous une liste (non exhaustive) de règles applicables : • Utiliser obligatoirement des outils permettant de minimiser les erreurs introduites durant le développement (outils d'analyse statique de code, utilisation de bibliothèques réputées pour leur sécurité, etc.). • Produire une documentation technique et dans le code décrivant l'implantation des protections développées (gestion de l'authentification, stockage des mots de passe, gestion des droits, chiffrement, etc.) ; • Respecter les normes de développement sécurisé, qu'elles soient propres au développeur, publiques ou propres à l'ACOSS ;

- S'obliger à corriger, au titre de la garantie et dans un temps raisonnable, les vulnérabilités introduites durant le développement qui lui sont remontées, en incluant automatiquement les corrections des autres occurrences des mêmes erreurs de programmation.
- Assurer un contrôle rigoureux des données d'entrées utilisateurs ;
- Sécuriser les accès aux fonctions d'administration ;
- Respecter le principe du moindre privilège ;
- Proscrire les mots de passe en dur dans le code ;
- Mettre en œuvre d'une gestion efficace des erreurs.

Pour la mise en œuvre de technologies web, les développements s'appuieront sur les recommandations de l'OWASP (Open Web Application Security Project) et notamment l'ASVS.

2.1.1.1.2.7 Transverse - Utilisation des templates FullStack

Étiquette du champ	Valeur du champ
ID de l'exigence	963
Nom	Transverse - Utilisation des templates FullStack
Etat exigence	47
Description	Les dépôts des différents projets (composants applicatifs, gestion des bundle archimed, construction et déploiement des chart helm) doivent être créés à partir des projets fullstack disponibles dans https://gitlab.altair.recouv/sdat/act/saul/fullstack. La version applicable est à minima le dernier tag de version X.Y.Z du dépôt fullstack au début du projet. Tous les fichiers inutiles pour le projet doivent être supprimés. Les projets fullstack sont des starter kit contenant des exemples d'api, de tests, de fichiers de configuration applicables sur les différentes plateformes cible. Les sources fullstack peuvent être modifiées pour répondre à des exigences techniques et/ou fonctionnelles non incluses dans ces exemples.

2.1.1.1.2.8 Transverse - Environnement d'exécution sur le poste de développement

Étiquette du champ	Valeur du champ
ID de l'exigence	876
Nom	Transverse - Environnement d'exécution sur le poste de développement
Etat exigence	39
Description	Chaque composant applicatif doit fournir un fichier "docker compose" permettant d'exécuter un ensemble cohérent (permettant de tester l'application) d'images sur le poste de développement.

2.1.1.1.3 Kubernetes

Étiquette du champ	Valeur du champ
ID de l'exigence	3213

Nom	Kubernetes
Etat exigence	4
Description	

2.1.1.1.3.1 Chart Helm - Sondes

Étiquette du champ	Valeur du champ
ID de l'exigence	2556
Nom	Chart Helm - Sondes
Etat exigence	13
Description	Tous les pods doivent être configurés avec des sondes liveness et readiness Celles ci pointent vers actuator dans le cas d'appli Spring ou d'autres mécanismes de vérification du bon fonctionnement sau besoin. Si nécessaire pour gérer des cas particuliers (de temps de démarrage long, préchauffage de donnée ou erreur spécifiques au démarrage,) une startup probe doit aussi être mise en place.

2.1.1.2 Gestion de configuration

Étiquette du champ	Valeur du champ
ID de l'exigence	950
Nom	Gestion de configuration
Etat exigence	6
Description	

2.1.1.2.1 Dépôts

Étiquette du champ	Valeur du champ
ID de l'exigence	951
Nom	Dépôts
Etat exigence	19
Description	Ceci est un test d'injection d'une exigence

2.1.1.2.1.1 Dépôt - Nomenclature des noms de dépôt

Étiquette du champ	Valeur du champ
ID de l'exigence	952
Nom	Dépôt - Nomenclature des noms de dépôt
Etat exigence	55
Description	Le nom des dépôts GIT doit être compatible avec l'expression régulière suivante: [a-z0-9-]*

2.1.1.2.1.2 Dépot - Groupe Gitlab

Étiquette du champ	Valeur du champ
ID de l'exigence	953
Nom	Dépot - Groupe Gitlab
Etat exigence	39
Description	Les dépôts GIT créés doivent se trouver dans un groupe préalablement fourni par l'URSSAF

2.1.1.2.1.3 Dépot - Création du dépôt pour les prototypes

Étiquette du champ	Valeur du champ
ID de l'exigence	954
Nom	Dépot - Création du dépôt pour les prototypes
Etat exigence	25
Description	Les dépôts temporaires, de prototypage doivent être créés dans un sous groupe "Bac à sable" fourni par l'URSSAF.

2.1.1.2.1.4 Dépot/Bouchon - Création des dépôts pour les bouchons

Étiquette du champ	Valeur du champ
ID de l'exigence	955
Nom	Dépot/Bouchon - Création des dépôts pour les bouchons
Etat exigence	30
Description	Les bouchons doivent être créés dans un sous groupe fourni par l'URSSAF.

2.1.1.2.1.5 Dépot/Backend - Création des dépôts pour composant Backend

Étiquette du champ	Valeur du champ
ID de l'exigence	956
Nom	Dépot/Backend - Création des dépôts pour composant Backend
Etat exigence	123
Description	Pour chaque composant applicatif de type backend, un dépôt GIT doit être créé. Le nom du dépôt doit suivre la nomenclature suivante: raf-<projet/appli>-<type>-<fonction> où type = [api, agt, lib, batch, bouchon] projet/appli=[recours, odp, optimdelai, grs, plat (platineum), rpa, sx (snv2)] fonction = [creance, controleur, parcours, profilage, ..., etc]

2.1.1.2.1.6 Dépot/FrontEnd - Création des dépôts pour composant Frontend

Étiquette du champ	Valeur du champ
ID de l'exigence	957
Nom	Dépôt/FrontEnd - Création des dépôts pour composant Frontend
Etat exigence	67
Description	Pour chaque composant applicatif de type frontend, un dépôt GIT doit être créé. Le nom du dépôt raf-<projet/appli>-ihm-<fonction> où projet/appli=[recours, odp, optimdelai, grs, plat (platineum), rpa, sx (snv2)] fonction = [creance, controleur, parcours, profilage, ..., etc]

2.1.1.2.1.7 Dépôt/Bundle Archimed - Création des dépôts de déploiement d'un bundle Archimed Backend

Étiquette du champ	Valeur du champ
ID de l'exigence	958
Nom	Dépôt/Bundle Archimed - Création des dépôts de déploiement d'un bundle Archimed Backend
Etat exigence	67
Description	Pour chaque bundle archimed back end, un dépôt GIT doit être créé. Son nom doit suivre la nomenclature suivante: raf-<nom projet>-bundle-archimed-backend Dans le cas ou un projet nécessite la création de plusieurs bundle backend. Un seul dépôt GIT doit être utilisé

2.1.1.2.1.8 Dépôt/Bundle Archimed - Création des dépôts de déploiement d'un bundle Archimed Frontend

Étiquette du champ	Valeur du champ
ID de l'exigence	959
Nom	Dépôt/Bundle Archimed - Création des dépôts de déploiement d'un bundle Archimed Frontend
Etat exigence	22
Description	Pour chaque bundle archimed front end, un dépôt GIT doit être créé. Son nom doit suivre la nomenclature suivante: raf-<nom projet>-bundle-archimed-frontend

2.1.1.2.1.9 Dépôt/Chart Helm - Création des dépôts de création du Chart Helm sur PFS

Étiquette du champ	Valeur du champ
ID de l'exigence	960
Nom	Dépôt/Chart Helm - Création des dépôts de création du Chart Helm sur PFS
Etat exigence	34
Description	Un dépôt GIT de création de chart helm doit être créé. Son nom doit suivre la nomenclature suivante: raf-<nom projet>-pfs-chart

2.1.1.2.1.10 Dépôt/Chart Helm - Création des dépôts de déploiement du Chart Helm

Étiquette du champ	Valeur du champ
ID de l'exigence	961
Nom	Dépôt/Chart Helm - Création des dépôts de déploiement du Chart Helm
Etat exigence	28
Description	Un dépôt GIT de création de déploiement sur la PFS doit être créé. Son nom doit suivre la nomenclature suivante: raf-<nom projet>-pfs-deploy Le dépôt comporte ensuite une branche par environnement selon la nomenclature suivante : • DEV/01 • DEV/02 • INT/01 • INT/02 • ... Sur la branche master du dépôt il ne doit y pas y avoir de fichier value.yaml ni env.sh mais leur équivalent finissant par .tmpl sans valorisation Le fichier namespace.yaml doit lui être rempli. Les tag lors des livraisons doivent être positionnés sur la branche master.

2.1.1.2.1.11 Dépôt - Création d'un dépôt documentaire

Étiquette du champ	Valeur du champ
ID de l'exigence	962
Nom	Dépôt - Création d'un dépôt documentaire
Etat exigence	33
Description	Un dépôt GIT documentaire doit être créé. Son nom doit suivre la nomenclature suivante: raf-<nom projet>-documentation Note: Ce dépôt contient les STD transverses, le manuel Dev&Int

2.1.1.2.2 Modèle de branchement

Étiquette du champ	Valeur du champ
ID de l'exigence	964
Nom	Modèle de branchement
Etat exigence	12
Description	

2.1.1.2.2.1 Modèle de branchement - Traçabilité des tâches

Étiquette du champ	Valeur du champ
ID de l'exigence	965
Nom	Modèle de branchement - Traçabilité des tâches
Etat exigence	54
Description	Chaque modification de SI doit être identifiée par un ticket d'anomalie ou d'évolution. Les tickets en visibilité du SDIT doivent faire l'objet d'une fiche de consignes GLD (Guichet de Livraison Documentaire) La fiche GLD doit être initialisée à partir du ticket Redmine de référence dans l'application : http://gld.urssaf.recouv/gld Les champs « description » et « intervention » doivent être complétés dans GLD si nécessaire Les tickets en visibilité du SDIT « corrigés par » doivent également faire l'objet d'une fiche de consigne GLD. Remarque Les tickets internes ne font pas l'objet de fiche GLD

2.1.1.2.2.2 Modèle de branchement - Messages de commit

Étiquette du champ	Valeur du champ
ID de l'exigence	966
Nom	Modèle de branchement - Messages de commit
Etat exigence	46
Description	Les messages des commit doivent suivre la nomenclature suivante: <#numéro du ticket redmine> : <code exigence si existant>-<descriptif de ce qui a été commité> ex: #12503 : message de commit

2.1.1.2.2.3 Modèle de branchement - Création des branches de travail

Étiquette du champ	Valeur du champ
ID de l'exigence	967
Nom	Modèle de branchement - Création des branches de travail
Etat exigence	15
Description	Chaque tâche de développement (fonctionnalité, correctif) doit être réalisée sur une branche GIT spécifique.

2.1.1.2.2.4 Modèle de branchement - Nommage des branches de travail

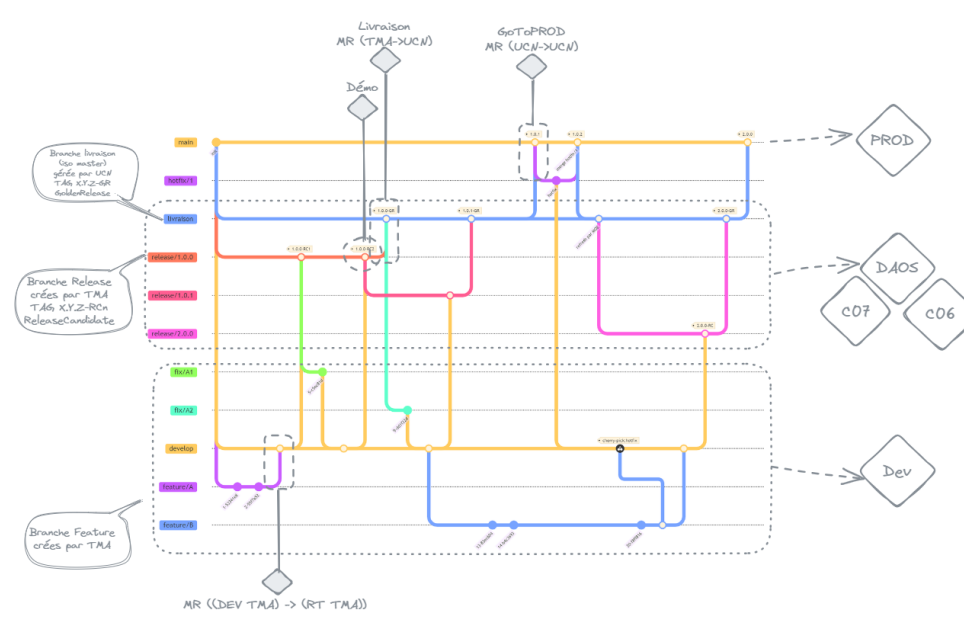
Étiquette du champ	Valeur du champ
ID de l'exigence	968
Nom	Modèle de branchement - Nommage des branches de travail
Etat exigence	19
Description	Le nom des branches de travail doivent suivre la nomenclature suivante: "feature/<Numéro tâche, ex: idRedmin>-<intitulé>" dans le cas de l'implémentation d'une nouvelle fonctionnalité ou d'un bug ne nécessitant pas un patch "hotfix/<Numéro tâche, ex: idRedmin>-<intitulé>" dans le cas de l'implémentation d'un correctif nécessitant la livraison d'un patch

Note: Ces règles de nommages sont imposées par fullStack

2.1.1.2.2.5 Modèle de branchement - Rapatriement d'un travail sur develop

Étiquette du champ	Valeur du champ
ID de l'exigence	969
Nom	Modèle de branchement - Rapatriement d'un travail sur develop
Etat exigence	15
Description	Les branches de travail sont mergées sur la branche de develop en utilisant le mécanisme de merge request et uniquement si les points de contrôles suivants sont vérifiés: • L'exécution de la pipeline associée à la branche est un succès • Au moins une approbation par un développeur tiers • Tous les commentaires ajoutés par les différents relecteurs sont résolus Justification: Le contrôle croisé par des pairs permet d'assurer la qualité du code, la montée en compétence de l'équipe de développement, ainsi qu'une homogénéisation du "style" de codage.

2.1.1.2.2.6 Modèle de branchement - GitFlow

Étiquette du champ	Valeur du champ
ID de l'exigence	2552
Nom	Modèle de branchement - GitFlow
Etat exigence	37
Description	L'utilisation de Git doit suivre le workflow suivant.  The diagram illustrates the GitFlow workflow. It shows a hierarchy of branches: main (production-ready), develop (integration branch), feature (for new features), release (for production releases), and hotfix (for quick response to production issues). Merge requests (MR) are used to move code between branches. Key MRs include: MR (TMA->UCN) for production releases, MR (UCN->UCN) for production releases, MR ((DEV TMA) -> (RT TMA)) for development, and MR (UCN->UCN) for production releases. The diagram also shows the flow of code from main to QAOS and PROD, and from develop to Dev.

Les livraisons doivent être préparées sur des branches release et livrées via une merge request vers la branche livraison assignée à UCN

Les tagss utilisés sur les branches release sont des tag release candidate "RC"

Les version acceptée seront taguées sur livraison par UCN avec des tag Golden Release "GR"

2.1.1.3 Comptes techniques

Étiquette du champ	Valeur du champ
ID de l'exigence	970
Nom	Comptes techniques
Etat exigence	8
Description	

2.1.1.3.1 Compte technique - Déclenchement procédure CADRAN

Étiquette du champ	Valeur du champ
ID de l'exigence	972
Nom	Compte technique - Déclenchement procédure CADRAN
Etat exigence	14
Description	Le fabricant doit demander le déclenchement de la procédure CADRAN dès le début des développements des composants frontend Le modèle de déclaration de droits métiers a utiliser est le document "Fiche cadran - droits métier.pdf" ----- http://gitlab.altair.recouv/sded/coeur-de-metier-recouvrement/controle-et-recouvrement-amiable-force-parametres/raf/raf-referentiel-projet/raf-document-techniques-applicables/-/raw/1.6.0/Documents%20Techniques%20Applicables/Authentication/Fiche%20cadran%20-%20droits%20m%C3%A9tier.pdf

2.1.1.3.2 Compte technique - Accès au portail technique

Étiquette du champ	Valeur du champ
ID de l'exigence	973
Nom	Compte technique - Accès au portail technique
Etat exigence	11
Description	Le fabricant doit demander les droits d'accès à la PFS de developpement. Note: Ce besoin apparaît lors de la phase de préintégration avant la livraison.

2.1.1.3.3 Compte technique/PFS - Accès à la PFS de developpement

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	974
Nom	Compte technique/PFS - Accès à la PFS de developpement
Etat exigence	11
Description	Le fabricant doit demander les droits d'accès au Portail Technique plateforme d'assemblage Archimed Note: Ce besoin apparaît lors de la phase de préintégration avant la livraison.

2.1.2 Contraintes de Construction

Étiquette du champ	Valeur du champ
ID de l'exigence	975
Nom	Contraintes de Construction
Etat exigence	10
Description	

2.1.2.1 Documents applicables

Étiquette du champ	Valeur du champ
ID de l'exigence	976
Nom	Documents applicables
Etat exigence	23
Description	Ceci est un test d'injection d'une exigence

2.1.2.2 Artefacts construits

Étiquette du champ	Valeur du champ
ID de l'exigence	993
Nom	Artefacts construits
Etat exigence	6
Description	

2.1.2.2.1 Artefact Maven

Étiquette du champ	Valeur du champ
ID de l'exigence	994
Nom	Artefact Maven
Etat exigence	10
Description	Ceci est un test d'injection d'une exigence

2.1.2.2.1.1 Artefact Maven - Nommage artifactId

Étiquette du champ	Valeur du champ
ID de l'exigence	995

Nom	Artefact Maven - Nommage artifactId
Etat exigence	47
Description	L'artifactId doit être identique au nom du dépôt GIT
Justification:	Cela permet d'assurer une homogénéisation des noms

2.1.2.2.1.2 Artefact Maven - Contract de service exposé

Étiquette du champ	Valeur du champ
ID de l'exigence	996
Nom	Artefact Maven - Contract de service exposé
Etat exigence	28
Description	Chaque contrat de service d'API exposée par un composant doit être déployé sur Nexus dans un artefact maven spécifique. Note: Chaque nouvelle version du fichier de contrat est dans le dépôt maven sous forme d'un artefact maven dédié. Note: Utiliser le goal <code>attach-artifact</code> du plugin <code>build-helper-maven-plugin</code> pour créer un artifact contenant le fichier descriptif. Le nom de l'artefact sera : <code><artifactId>-<classifier>.<extension></code> classifier={<code>swagger</code>,<code>soap</code>} extension={<code>yaml</code>,<code>json</code>,<code>wsdl</code>,<code>xml</code>} Justification: Une fois déployés sur le dépôt maven, ces fichiers peuvent être intégrés automatiquement dans les composants consommateurs.

2.1.2.2.1.3 Artefact Maven - Caractéristiques

Étiquette du champ	Valeur du champ
ID de l'exigence	997
Nom	Artefact Maven - Caractéristiques
Etat exigence	126
Description	Les binaires doivent être relasés et mis à disposition dans Nexus avec les caractéristiques suivantes : GroupId : <code>fr.acoss.recouv.raf.<fonction>.<sous fonction></code> ArtifactId: <code>.<nom_module></code> Version : <code>X.Y.Z</code> Ex : <code>http://nexus.altair.recouv/nexus/content/repository/releases/fr/acoss/recouv/raf/gestionnaire/ihm/raf-gestionnaire-ihm-1.0.1.tar.gz</code>

2.1.2.2.1.4 Configuration - Construction artefact maven

Étiquette du champ	Valeur du champ
ID de l'exigence	998
Nom	Configuration - Construction artefact maven
Etat exigence	8
Description	

Chaque composant applicatif contient dans ses sources le fichier permettant la construction de ses artefacts maven (le pom.xml)

2.1.2.2.2 Bundle Archimed

Étiquette du champ	Valeur du champ
ID de l'exigence	999
Nom	Bundle Archimed
Etat exigence	14
Description	

2.1.2.2.2.1 Bundle Archimed - Cohérence contract/règles de sécurité

Étiquette du champ	Valeur du champ
ID de l'exigence	1000
Nom	Bundle Archimed - Cohérence contract/règles de sécurité
Etat exigence	28
Description	La cohérence entre l'API et le bundle d'exposition sur Archimed doit être garantie par l'ESN Titulaire. Pour chaque opération défini dans le contrat, une règle de sécurité doit lui être associée. Note: Tout ajout/suppression/renommage d'opération exposée nécessite de mettre à jour les règles

2.1.2.2.2.2 Bundle Archimed - Validation des règles de sécurité

Étiquette du champ	Valeur du champ
ID de l'exigence	1001
Nom	Bundle Archimed - Validation des règles de sécurité
Etat exigence	8
Description	La liste des API décrites dans un bundle Archimed et les règles de sécurité doivent être validées par l'architecture d'entreprise et la sécurité. Justification: Plusieurs composants peuvent être décrit dans un même bundle. L'exception à cette règle est que si un composant est client d'une API externe, alors il faut envisager un bundle dédié. Le découpage logique doit être validé par l'architecte.

2.1.2.2.2.3 Bundle Archimed - Localisation des contrats des services

Étiquette du champ	Valeur du champ
ID de l'exigence	1002
Nom	Bundle Archimed - Localisation des contrats des services
Etat exigence	58
Description	Les contrats de service présents sur Nexus ne doivent pas faire partie des sources du bundle. Ils doivent être récupérés automatiquement lors de la phase de construction du bundle.

2.1.2.2.2.4 Bundle Archimed - Nommage d'un bundle backend

Étiquette du champ	Valeur du champ
ID de l'exigence	1003
Nom	Bundle Archimed - Nommage d'un bundle backend
Etat exigence	39
Description	Le nom d'un bundle backend doit suivre la nomenclature suivante: "<nom du projet en majuscule>-BE"

2.1.2.2.2.5 Bundle Archimed - Structure d'un bundle backend

Étiquette du champ	Valeur du champ
ID de l'exigence	1004
Nom	Bundle Archimed - Structure d'un bundle backend
Etat exigence	24
Description	Un bundle backend est une archive devant posséder la structure suivante: - regles_nom du bundle>_<version>.json ---> Déclaration des règles de routages - rest_<nom du bundle>_<nom contract>_<version>.yaml ---> Un fichier par contract swagger de type rest - soap_<nom du bundle>_<nom contract>_<version>.xml ---> Un fichier par contract de type soap Note : Le fichier décrivant la construction du bundle est présent dans les sources de celui-ci mais est absent de l'archive.

2.1.2.2.2.6 Bundle Archimed - Nommage d'un bundle frontend

Étiquette du champ	Valeur du champ
ID de l'exigence	1005
Nom	Bundle Archimed - Nommage d'un bundle frontend
Etat exigence	22
Description	Le nom d'un bundle frontend doit être le code MEGA

2.1.2.2.2.7 Bundle Archimed - Structure d'un bundle frontend

Étiquette du champ	Valeur du champ
ID de l'exigence	1006
Nom	Bundle Archimed - Structure d'un bundle frontend
Etat exigence	8
Description	Un bundle frontend est une archive devant posséder à minima la structure suivante: conf_frontend-<nom du bundle>_<version>.json ---> Fichier décrivant la configuration (sécurité, contacts) Note : Le fichier décrivant la construction du bundle est présent dans les sources de celui-ci mais est absent de l'archive.

2.1.2.2.2.8 Bundle Archimed - Métainformations à ajouter aux swaggers

Étiquette du champ	Valeur du champ
ID de l'exigence	1007
Nom	Bundle Archimed - Métainformations à ajouter aux swaggers
Etat exigence	24
Description	Les swaggers exposés par Archimed doivent contenir à minima: • L'extension "x-backendArchimed" dans la section "info" qui a pour valeur le code de votre application, qui a été renseigné dans la grille d'autorisation Prisme • Si le contrat contient un champ "schemas" (Swagger 2.0), le tableau doit contenir la valeur "http" ou bien le contrat ne doit pas contenir de champ "schemas" • Toutes les opérations du contrat doivent avoir: ○ Une valeur pour le champ "operationId", même si le service est non-sécurisé. Il est nécessaire d'avoir une valeur unique pour chaque opération, même pour des contrats différents ○ Le tag "archimed-double-exposition" doit être positionné afin de pouvoir faire des tests non sécurisé. Ce tag déclenche la proxification des services exposés sur la plateforme d'assemblage. Deux path d'entrée sont exposés: un sécurisé et un non sécurisé pour les tests d'intégration • Le chemin de base d'accès aux services, contenu dans le champ "basePath" doit avoir la forme suivante : /<domaine>/<version>/<chemin> • Le <domaine> peut être simple (exemple : "archdemo") ou composé (exemple : "archimed/demo") ○ La <version> doit être de la forme vX.Y (exemples : v1, v1.2.3, v2.10) ○ Le <chemin> est facultatif, et peut être simple ou composé Exemples : /mon_appli/v1 ; /mon/appli/v1/services/rest ; /app/archimed/v1.1/mes_proxy Justification: Archimed se base sur les operationsId et le x-backendArchimed pour configurer la Gaetawy et l'application des règles

2.1.2.2.2.9 Bundle Archimed/SNV2 - Métainformations à ajouter aux swaggers

Étiquette du champ	Valeur du champ
ID de l'exigence	1008
Nom	Bundle Archimed/SNV2 - Métainformations à ajouter aux swaggers
Etat exigence	22
Description	Les swaggers exposés par Archimed dans l'environnement SNV2 doivent contenir à minima: • dans le bloc "info", les propriétés: • x-typeBackend: "conteneurK8S" • x-serviceConteneur: "<nom-image-container-sx>" sans la base cible (Le nom de l'image est présent dans le fichier ws-java-deploy.json des machines SX) • pour chaque operation, il faut définir la propriété "x-baseSrv2". Les valeurs possibles sont "p" pour base de mise à jour et "c" pour base consultation post batch

2.1.2.2.2.10 Bundle Archimed/PFS - Métainformations à ajouter aux swaggers

Étiquette du champ	Valeur du champ
ID de l'exigence	1009
Nom	Bundle Archimed/PFS - Métainformations à ajouter aux swaggers

Etat exigence	22
Description	Les swaggers exposés par Archimed dans l'environnement PFS doivent contenir à minima: • dans le bloc "info", les propriétés: • x-typeBackend: "interne"

2.1.2.2.2.11 Bundle Archimed - Version

Étiquette du champ	Valeur du champ
ID de l'exigence	1798
Nom	Bundle Archimed - Version
Etat exigence	45
Description	Le dépôt git du bundle doit être taggué avec la même version que celle de la configuration qui sera installée dans Archimed.

2.1.2.2.2.12 Bundle Archimed - Règles de sécurité

Étiquette du champ	Valeur du champ
ID de l'exigence	1803
Nom	Bundle Archimed - Règles de sécurité
Etat exigence	11
Description	Les appels issues d'une IHM doivent porter les droits de l'utilisateur le plus loin possible lorsque le backend fait des appels complémentaires à d'autres backend. Dans les cas ou ce n'est pas possible : MOE tierce, Batch, un scope dédié représentant le composant technique doit être déclaré dans le bundle.

2.1.2.2.3 Chart Helm (PFS)

Étiquette du champ	Valeur du champ
ID de l'exigence	1010
Nom	Chart Helm (PFS)
Etat exigence	19
Description	

2.1.2.2.3.1 Chart Helm - Structure

Étiquette du champ	Valeur du champ
ID de l'exigence	1011
Nom	Chart Helm - Structure
Etat exigence	31
Description	

Un chart helm est une archive dont l'arborescence interne doit être à minima la suivante:

- config/ ---> Fichiers de configuration utilisés dans l'environnement de la PFS
- security/ ---> Keystores utilisés par prisme templates/
- cronjob/ ---> Template des manifests de description des cronjob
- deployments/ ---> Template des manifests décrivant les pod
- service/ ---> Template des manifests des services au sens K8s
- ingress/ ---> Template de la configuration ingress
- Chart.yaml ---> Descriptif du chart helm contenant la version et le nom
- application.yaml ---> Fichier contenant les variable helm statiques indépendantes de l'environnement cible (ex: identifiant image)
- values.yaml ---> Fichier contenant les variable helm dépendant l'environnement cible (ex: quotas, url)

Note: Les templates sont analysés par helm qui substitue les variables helm par des valeurs

2.1.2.2.3.2 Chart Helm - Nommage du chart

Étiquette du champ	Valeur du champ
ID de l'exigence	1012
Nom	Chart Helm - Nommage du chart
Etat exigence	25
Description	
Le nom d'un chart helm doit suivre la nomenclature "charts-<nom projet>"	

2.1.2.2.3.3 Chart Helm - Version

Étiquette du champ	Valeur du champ
ID de l'exigence	1013
Nom	Chart Helm - Version
Etat exigence	8
Description	
La version d'un chart helm version doit être incrémentée à chaque modification du chart tout en étant cohérente avec la version de l'application.	
Note: Suivre la nomenclature suivante : https://semver.org/	

2.1.2.2.3.4 Chart Helm - Variabilisation des templates

Étiquette du champ	Valeur du champ
ID de l'exigence	1014
Nom	Chart Helm - Variabilisation des templates
Etat exigence	28
Description	
Les templates ne doivent pas contenir de valeurs liées aux environnements d'exécution en dur.	
Ces valeurs doivent être déclarées via des variables helm.	
Les variables créées dans les fichiers values.yaml doivent être accompagnées de commentaire donnant des indications sur les valeurs attendues (ex : valeur utilisée en dev)	

Le value.yaml du chart doit comprendre toutes les paramètres du chart et les valeurs par défaut des subchart qu'il utilise si celles-ci ne sont pas à adapter selon les environnements.

2.1.2.2.3.5 Chart Helm - Variabilisation des fichiers de configuration des composants

Étiquette du champ	Valeur du champ
ID de l'exigence	1015
Nom	Chart Helm - Variabilisation des fichiers de configuration des composants
Etat exigence	25
Description	Tous les paramètres de configuration d'un composant liés à l'environnement d'exécution doivent être variabilisés par des variables helm. Note: Cela se traduit entre autre par la présence de variables helm dans les fichiers de configuration.

2.1.2.2.3.6 Chart Helm - Base de données

Étiquette du champ	Valeur du champ
ID de l'exigence	1016
Nom	Chart Helm - Base de données
Etat exigence	27
Description	La déclaration d'un cluster de base de données doit se faire en utilisant le subchart URSSAF "postgres" Note: https://readthedocs.cnp.recouv/docs/documentation-pfs/fr/latest/composants/operateur-postgres/index.html

2.1.2.2.3.7 Chart Helm - Volumétrie

Étiquette du champ	Valeur du champ
ID de l'exigence	1017
Nom	Chart Helm - Volumétrie
Etat exigence	8
Description	Un chart helm doit avoir une taille inférieure à 1Mo Justification: Les chart helm sont stockés comme des secrets dans K8s, or K8s impose une limite quant à la taille des secrets.

2.1.2.2.3.8 Chart Helm - Exposition des services

Étiquette du champ	Valeur du champ
ID de l'exigence	1018
Nom	Chart Helm - Exposition des services
Etat exigence	26
Description	La configuration ingress ne doit déclarer qu'un host par bundle Archimed. Si un bundle contient plusieurs microservices, le mécanisme de routage doit utiliser l'attribut "path" doit être utilisé. L'url d'accès à un microservice suit alors la nomenclature suivante :

ex:
 <host>/<nom application>/<element discriminant vers le backend>
 host : Url de la Gateway
 nom de l'application: nom du bundle sans le -BE
 element discriminant vers le backend : element discriminant du micro service

2.1.2.2.3.9 Chart Helm - Configurabilité de la sécurité

Étiquette du champ	Valeur du champ
ID de l'exigence	1019
Nom	Chart Helm - Configurabilité de la sécurité
Etat exigence	25
Description	En utilisant le mécanisme de variabilisation Helm, il doit être possible de désactiver/activer Prisme sur chacun des composants applicatifs. Justification: Les premiers tests d'intégration s'effectuent sans activer la sécurisation

2.1.2.2.3.10 Chart Helm - Configurabilité des ressources

Étiquette du champ	Valeur du champ
ID de l'exigence	1020
Nom	Chart Helm - Configurabilité des ressources
Etat exigence	25
Description	En utilisant le mécanisme de variabilisation Helm, il doit être possible de configurer pour chaque container les propriétés suivantes: • limits memory • limits cpu • requests memory • requests cpu • Les caractéristique de déploiement des item kubernetes (pod, job, conrkob) tel que le nombre de réplicas Note: Ces valeurs doivent être cohérentes avec les besoin de l'applicatif sous jacent. Si les ressources ne sont pas suffisantes, il faut faire valider les valeurs par l'architecte

2.1.2.2.3.11 Chart Helm - Configurabilité d'accès aux services externes

Étiquette du champ	Valeur du champ
ID de l'exigence	1021
Nom	Chart Helm - Configurabilité d'accès aux services externes
Etat exigence	25
Description	En utilisant le mécanisme de variabilisation Helm, il doit être possible de configurer à minima pour chaque composant applicatif: • Les URL manipulées par ingress • Les informations d'accès aux API consommées • Les informations de sécurité liées à prisme, aux cors

2.1.2.2.3.12 Chart Helm - Gestion mots de passe

Étiquette du champ	Valeur du champ
ID de l'exigence	1022
Nom	Chart Helm - Gestion mots de passe
Etat exigence	26
Description	L'outil Vault doit être utilisé pour stocker toutes les variables nécessitant d'être chiffrées. La gestion des mots de passe dans un environnement kubernetes doit donc utiliser le coffre-fort à secrets "Vault". Note: Des exemples d'initialisation de services PFS utilisant Vault peuvent être fournis par l'équipe responsable du socle.

2.1.2.2.3.13 Chart Helm - Déploiement

Étiquette du champ	Valeur du champ
ID de l'exigence	1023
Nom	Chart Helm - Déploiement
Etat exigence	10
Description	Le déploiement passe par un dépôt git « Deploy ». Il comporte une branche master qui sert de template pour la valorisation de base du value.yaml et namespace.yaml. Le value.yaml sur cette branche n'est pas rempli. La branche master doit être livrée en même temps que les nouvelles versions du chart et elle doit être taguée en conséquence. Le dépôt comporte ensuite une branche par environnemen selon la nomenclature suivante : • DEV/01 • DEV/02 • INT/01 • INT/07 La structure de ce dépôt est définie ici : https://gitlab.altair.recouv/sddos/innovation-et-transformation-digitale/procedure_deploiement_pfs/depotReference/-/tree/master

2.1.2.2.3.14 Chart Helm - Dépôt de stockage

Étiquette du champ	Valeur du champ
ID de l'exigence	1024
Nom	Chart Helm - Dépôt de stockage
Etat exigence	25
Description	Le registre Harbor est utilisé pour le stockage des charts HELM. Il existe un registry de dev et un de prod. L'utilisation du Jenkins est nécessaire pour builder et déposer les charts sur Harbor.

2.1.2.2.3.15 Chart Helm - Exposition de la base de donnée

Étiquette du champ	Valeur du champ
ID de l'exigence	1416
Nom	Chart Helm - Exposition de la base de donnée
Etat exigence	17
Description	Le chart helm doit contenir une ressource permettant d'exposer les bases de données à l'extérieur du cluster k8s. Cette ressource doit avoir les caractéristiques suivantes : • Elle n'est pas déployée sur l'environnement de production. • L'activation ou non du déploiement de cette ressource doit être configurable dans le fichier "values.yaml" du projet de déploiement. • Elle est de type "LoadBalancer"

2.1.2.2.3.16 Chart Helm - Polices de sécurité

Étiquette du champ	Valeur du champ
ID de l'exigence	1417
Nom	Chart Helm - Polices de sécurité
Etat exigence	17
Description	Le charhelm doit contenir une ressource de type "CiliumNetworkPolicy" déclarant les polices de sécurités régissant les flux entrants/sortants des ressources déployées. Les règles de sécurités minimales sont les suivantes : Les flux inter namespace directs sont interdits. Il est nécessaire de passer par Archimed et/ou un Bus Kafka Chaque flux intra namespace est autorisé. Un pod peut par exemple communiquer directement avec un autre pod du namespace sans passer par archimed. Chaque flux sortant et entrant de la PFS doit être identifié et faire l'objet d'une règle spécifique à faire valider par UCN.

2.1.2.2.3.17 Chart Helm - Operateurs et subcharts

Étiquette du champ	Valeur du champ
ID de l'exigence	1645
Nom	Chart Helm - Operateurs et subcharts
Etat exigence	11
Description	Les charts livrés doivent utiliser les dernière version des opérateurs et subcharts (postgres, s3) disponibles.

2.1.2.2.4 Image Docker

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	1025
Nom	Image Docker
Etat exigence	9
Description	

2.1.2.2.4.1 Image Docker - Nommage

Étiquette du champ	Valeur du champ
ID de l'exigence	1026
Nom	Image Docker - Nommage
Etat exigence	32
Description	Le nom d'une image docker doit contenir: Le nom du de depot git La version

2.1.2.2.4.2 Image Docker - Versioning

Étiquette du champ	Valeur du champ
ID de l'exigence	1027
Nom	Image Docker - Versioning
Etat exigence	27
Description	La version d'une image docker doit suivre celle du composant applicatif

2.1.2.2.4.3 Image Docker - Configurabilité

Étiquette du champ	Valeur du champ
ID de l'exigence	1028
Nom	Image Docker - Configurabilité
Etat exigence	8
Description	Chaque composant applicatif manipule des fichiers de configuration spécifiques à l'environnement d'exécution. Ils sont accessibles dans l'image via des volumes.

2.1.2.2.4.4 Image Docker - Environnement de développement

Étiquette du champ	Valeur du champ
ID de l'exigence	1029
Nom	Image Docker - Environnement de développement
Etat exigence	24
Description	Chaque composant applicatif doit fournir les fichiers dockerfiles permettant de construire des images exécutables sur l'environnement de développement.

Si le service consomme des API, alors un fichier docker-compose doit permettre de lancer toutes les images des API consommés.

Note: L'environnement de développement s'appelle "demo". il doit apparaître en suffixe du ,nom du dockerfile

2.1.2.2.4.5 Image Docker - Dépôt de stockage

Étiquette du champ	Valeur du champ
ID de l'exigence	1030
Nom	Image Docker - Dépôt de stockage
Etat exigence	230
Description	Le registre Harbor est utilisé pour le stockage des images. Il existe un registry de dev et un de prod. L'utilisation du Jenkins est nécessaire pour builder et déposer les images sur Harbor.

2.1.2.2.4.6 Configuration - Génération des Images Docker

Étiquette du champ	Valeur du champ
ID de l'exigence	1031
Nom	Configuration - Génération des Images Docker
Etat exigence	43
Description	Chaque composant applicatif contient dans ses sources les fichiers permettant la construction des images docker (les docker files): • l'image d'exécution de l'API • au besoin, une image pour les migrations de base de données (basée sur liquibase)

2.1.2.2.4.7 Image Docker - Configuration de la timezone

Étiquette du champ	Valeur du champ
ID de l'exigence	1032
Nom	Image Docker - Configuration de la timezone
Etat exigence	101
Description	Objectif de l'exigence : Initialiser la timezone avec la valeur adéquate lors de l'instanciation de l'image Cas d'utilisation : Exploitation des containers Fonctionnalité associée : Date / Time utilisé par les instances de container

2.1.2.3 Chaines de construction

Étiquette du champ	Valeur du champ
ID de l'exigence	1033

Nom	Chaines de construction
Etat exigence	5
Description	

2.1.2.3.1 Généralités

Étiquette du champ	Valeur du champ
ID de l'exigence	1034
Nom	Généralités
Etat exigence	18
Description	

2.1.2.3.1.1 Chaîne de construction - Utilisation de Jenkins

Étiquette du champ	Valeur du champ
ID de l'exigence	1035
Nom	Chaîne de construction - Utilisation de Jenkins
Etat exigence	38
Description	
La construction et le déploiement des artefacts (artefact maven, images docker, bundles archimed, chart helm) doit être déclenchable via des pipeline de construction Jenkins.	

2.1.2.3.1.2 Chaîne de construction - Fourniture de l'intégration par composant

Étiquette du champ	Valeur du champ
ID de l'exigence	1036
Nom	Chaîne de construction - Fourniture de l'intégration par composant
Etat exigence	31
Description	
Tout composant applicatif doit fournir l'intégration continue associée.	

2.1.2.3.1.3 Chaîne de construction - Construction des artefacts ACOSS

Étiquette du champ	Valeur du champ
ID de l'exigence	1037
Nom	Chaîne de construction - Construction des artefacts ACOSS
Etat exigence	8
Description	
Les artefacts contruits et déployés doivent être construits avec la plateforme d'intégration continue ACOSS.	

2.1.2.3.1.4 Chaîne de construction - Caractéristiques

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	1038
Nom	Chaine de construction - Caractéristiques
Etat exigence	56
Description	Les pipelines jenkins doivent être multibranches et doivent être configurées de manière à: • Se builder automatiquement après chaque push sur une branche • [Si le dépôt contient des branches] Détecter automatiquement l'ajout/suppression de branches • [Si le dépôt contient du code source] Lancer automatiquement la chaine de construction en cas de push

2.1.2.3.1.5 Chaine de construction - Notification en cas d'erreur

Étiquette du champ	Valeur du champ
ID de l'exigence	1039
Nom	Chaine de construction - Notification en cas d'erreur
Etat exigence	46
Description	Si une erreur survient lors de l'exécution d'une pipeline de construction, Alors: • La construction doit échouer • La pipeline doit être dans l'état FAILURE • Un message de log explicite doit être émis • Un mail doit être envoyé à une liste d'adresse mails fournies par l'URSSAF. Note: La liste des adresses mails doit être une chaîne de caractères qui est la concaténation des adresses mails cibles séparées par le caractère ','. Elle est stockée dans une variable d'environnement.

2.1.2.3.1.6 Chaine de construction - Variables d'environnement

Étiquette du champ	Valeur du champ
ID de l'exigence	1040
Nom	Chaine de construction - Variables d'environnement
Etat exigence	32
Description	Toute donnée contextuelle (URL, nom de dépôt, nom de projet) présente dans un jenkinsfile doit être stockées dans une variable d'environnement du JenkinsFile

2.1.2.3.1.7 Chaine de construction/Composant - Création/Déploiement composant applicatif

Étiquette du champ	Valeur du champ
ID de l'exigence	1041
Nom	Chaine de construction/Composant - Création/Déploiement composant applicatif
Etat exigence	180
Description	Chaque composant applicatifBackend/FrontEnd doit avoir sa propre pipeline de construction jenkins des composants applicatifs réalisant à minimum les tâches suivantes: • La compilation • L'exécution des tests unitaires inclus dans les sources.

- Générer un rapport de qualité
- Analyser les vulnérabilités (via Trivy et Dependency Check)
- Ajout des Tag GIT
- [Optionnel] La génération des images PFS
- Le déploiement des artifacts maven sur Nexus
- Le déploiement de l'image sur Harbor

2.1.2.3.1.8 Chaîne de construction/Bundle Archimed - Création/Installation Bundle Backend

Étiquette du champ	Valeur du champ
ID de l'exigence	1042
Nom	Chaîne de construction/Bundle Archimed - Création/Installation Bundle Backend
Etat exigence	1025
Description	Une pipeline doit être créée pour: • Construire le(s) bundle(s) archimed back end. • Installer le(s) bundle(s) sur Archimed

2.1.2.3.1.9 Chaîne de construction/Bundle Archimed - Création/Installation Bundle FrontEnd

Étiquette du champ	Valeur du champ
ID de l'exigence	1043
Nom	Chaîne de construction/Bundle Archimed - Création/Installation Bundle FrontEnd
Etat exigence	46
Description	Une pipeline doit être créée pour: • Construire le(s) bundle(s) archimed front end. • Installer le(s) bundle(s) sur Archimed

2.1.2.3.1.10 Chaîne de construction/Chart Helm - Création

Étiquette du champ	Valeur du champ
ID de l'exigence	1044
Nom	Chaîne de construction/Chart Helm - Création
Etat exigence	30
Description	Une pipeline doit être créée pour • construire un charthelm • le déployer sur Harbor • [OPTIONEL] pour chaque namespace PFS cible, permettre ou non de lancer automatiquement la pipeline de déploiement.

2.1.2.3.1.11 Chaîne de construction/Chart Helm - Déploiement sur environnement exécution

Étiquette du champ	Valeur du champ
ID de l'exigence	1045
Nom	Chaîne de construction/Chart Helm - Déploiement sur environnement exécution
Etat exigence	26
Description	Une pipeline doit être créée pour: - Enrichir le chart helm statiques avec les données spécifique à l'environnement d'exécution (ex: certificats de sécurité, fichier des données statiques (host, sécurité ...)) - Déployer le chart helm sur la PFS.

2.1.2.3.1.12 Configuration - Définition des chaînes de construction

Étiquette du champ	Valeur du champ
ID de l'exigence	1046
Nom	Configuration - Définition des chaînes de construction
Etat exigence	31
Description	Chaque composant applicatif contient dans ses sources les fichiers permettant d'intégrer la chaîne de construction et de déploiement dans la PIC: - jenkinsfile Note: Si besoin, plusieurs jenkinsfile peuvent être développés. Par exemple un jenkins file pour une construction locale et une pour la PIC URSSAF. - scripts

2.1.2.3.2 Sécurité

Étiquette du champ	Valeur du champ
ID de l'exigence	1047
Nom	Sécurité
Etat exigence	7
Description	

2.1.2.3.2.1 Chaîne de construction/Sécurité - Outil de contrôle des dépendances

Étiquette du champ	Valeur du champ
ID de l'exigence	1048
Nom	Chaîne de construction/Sécurité - Outil de contrôle des dépendances
Etat exigence	53
Description	L'outil "Dependency Check" de l'OWASP doit être mis en place dans les pipelines Jenkins afin de vérifier les vulnérabilités dans les bibliothèques externes utilisées par le projet.

2.1.2.3.2.2 Chaîne de construction/Sécurité - Seuil d'acceptabilité des violations de dépendances

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	1049
Nom	Chaine de construction/Sécurité - Seuil d'acceptabilité des violations de dépendances
Etat exigence	35
Description	Pour l'analyse des CVE seul le niveau de confiance "Highest" sera pris en compte. Le niveau maximum des vulnérabilités accepté est "haut". Toutes les vulnérabilités de niveau "Critique" avec un niveau de confiance "Highest" devront faire l'objet d'une demande auprès des chargés de conformité pour acceptation exceptionnelle.

2.1.2.3.2.3 Chaine de construction/Sécurité - Outil d'analyse des vulnérabilité

Étiquette du champ	Valeur du champ
ID de l'exigence	1050
Nom	Chaine de construction/Sécurité - Outil d'analyse des vulnérabilité
Etat exigence	25
Description	L'outil Trivy de recherche de vulnérabilité dans les images de container doit être mis en place dans les pipelines Jenkins

2.1.2.3.2.4 Chaine de construction/Sécurité - Seuil d'acceptabilité des violations des vulnérabilités

Étiquette du champ	Valeur du champ
ID de l'exigence	1051
Nom	Chaine de construction/Sécurité - Seuil d'acceptabilité des violations des vulnérabilités
Etat exigence	30
Description	Le niveau maximum des vulnérabilités accepté est "haut". Toutes les vulnérabilités de niveau "Critique" doivent faire l'objet d'une demande auprès des chargés de conformité pour acceptation exceptionnelle.

2.1.2.3.2.5 Chaine de construction/Sécurité - Mise à disposition du rapport de vulnérabilité

Étiquette du champ	Valeur du champ
ID de l'exigence	1052
Nom	Chaine de construction/Sécurité - Mise à disposition du rapport de vulnérabilité
Etat exigence	30
Description	Le rapport de vulnérabilité produit par les pipeline de construction de composant applicatif doivent être consultable

2.1.2.3.3 Qualité

Étiquette du champ	Valeur du champ
ID de l'exigence	1053

Nom	Qualité
Etat exigence	9
Description	

2.1.2.3.3.1 Chaîne de construction/Composant - Barrière qualité

Étiquette du champ	Valeur du champ
ID de l'exigence	1054
Nom	Chaîne de construction/Composant - Barrière qualité
Etat exigence	13
Description	Le code doit respecter le niveau de qualité les seuils de contrôles associés à ces métriques sont définies au niveaux des profils SonarQube URSSAF. La barrière qualité dans SonarQube permet de vérifier que le code analysé respecte les exigences en termes de qualité : Respect des règles mais aussi de certaines métriques (pourcentage de commentaires, pourcentage de tests unitaires, etc...). Elle n'est pas spécifique à un langage de programmation seul le mode de calcul de ses métriques est différent (Profil qualité). Les métriques vérifiées par la barrière qualité sont : • Pourcentage de code couvert par les tests unitaires • Pourcentage de lignes de code dupliquées • Score de maintenabilité du code • Score de fiabilité du code • Score de sécurité du code • Pourcentage de Security Hotspots pris en compte

2.1.2.3.3.2 Chaîne de construction/Composant - Violations de qualités

Étiquette du champ	Valeur du champ
ID de l'exigence	1055
Nom	Chaîne de construction/Composant - Violations de qualités
Etat exigence	26
Description	Si des violations qualités dans le code source sont détectées, Alors une erreur doit être émise.

2.1.2.3.3.3 Chaîne de construction/Composant - Mise à disposition du rapport qualité

Étiquette du champ	Valeur du champ
ID de l'exigence	1056
Nom	Chaîne de construction/Composant - Mise à disposition du rapport qualité
Etat exigence	45
Description	Le rapport qualité produit par les pipeline de construction de composant applicatif doivent être déployés sur SonarQube

2.1.2.3.3.4 Qualité - Règles de qualité TypeScript

Étiquette du champ	Valeur du champ
ID de l'exigence	1057
Nom	Qualité - Règles de qualité TypeScript
Etat exigence	25
Description	Les règles de qualité TYPESCRIPT à appliquer sont celles définies dans un fichier associé au profil Typescript Sonar way dans le Sonarqube de la nouvelle PIC UCN. La liste des règles est mises à jour avec les montées de version de Sonarqube. Note: La procédure pour exporter les règles sonar est décrite dans : http://gitlab.altair.recouv/bt/outils/pic/-/wikis/Sonarqube/Exporter-Profil-Qualit%C3%A9

2.1.2.3.3.5 Qualité - Règles de qualités Java

Étiquette du champ	Valeur du champ
ID de l'exigence	1058
Nom	Qualité - Règles de qualités Java
Etat exigence	28
Description	Les règles de qualité JAVA à appliquer sont celles définies dans le fichier associé au profil Java Sonar way dans le Sonarqube de la nouvelle PIC UCN. La liste des règles est mise à jour avec les montées de version de Sonarqube Note: La procédure pour exporter les règles sonar est décrite dans : http://gitlab.altair.recouv/bt/outils/pic/-/wikis/Sonarqube/Exporter-Profil-Qualit%C3%A9

2.1.2.3.3.6 Qualité - Seuils autorisés

Étiquette du champ	Valeur du champ
ID de l'exigence	1059
Nom	Qualité - Seuils autorisés
Etat exigence	9
Description	Les seuils à atteindre sont ceux définis par la barrière de qualité URSSAF intégrées dans les chaînes de construction. Toute violation doit être justifiée et approuvée par l'URSSAF.

2.1.2.3.3.7 Qualité - Vérification des scripts shell

Étiquette du champ	Valeur du champ
ID de l'exigence	1619
Nom	Qualité - Vérification des scripts shell
Etat exigence	14
Description	

Tous Shell (/bash, /zsh/, /ksh...) livré (Requête technique ou batch), pour être valable doit passer le test du ShellCheck sans erreur, ni notice, ni warning, ni info.

L'objectif est de sécuriser les opérations sur les environnements et de gagner en qualité.

Plusieurs possibilités pour passer le ShellCheck :

En ligne de commande

Plugin sur les IDEs

L'outil propose des corrections et des explications suite à l'analyse qui est faite.

Attention, pour de raison de sécurité, il ne faut pas utiliser la version web proposé par l'éditeur.

<http://confluence.urssaf.recouv/display/ENFAPORT/ENFp+-+Valider+vos+script+Shell+avec+ShellCheck>

2.1.3 Contraintes de Livraison

Étiquette du champ	Valeur du champ
ID de l'exigence	1060
Nom	Contraintes de Livraison
Etat exigence	11
Description	

2.1.3.1 Documents applicables

Étiquette du champ	Valeur du champ
ID de l'exigence	1061
Nom	Documents applicables
Etat exigence	13
Description	
Ceci est un test d'injection d'une exigence	

2.1.3.2 Livrables

Étiquette du champ	Valeur du champ
ID de l'exigence	1065
Nom	Livrables
Etat exigence	26
Description	

2.1.3.2.1 Livrable - Rapport des tests usines

Étiquette du champ	Valeur du champ
ID de l'exigence	1066
Nom	Livrable - Rapport des tests usines

Etat exigence	17
Description	
Le rapport rendu de tests doit contenir: L'état d'exécution des test réalisés en recette usine Les indicateurs de qualités (qualité de code et couverture de tests) Les justifications si les seuils attendus sur les différents indicateurs ne sont pas atteints	

2.1.3.2.2 Livrable - Procédure de migration

Étiquette du champ	Valeur du champ
ID de l'exigence	1068
Nom	Livrable - Procédure de migration
Etat exigence	18
Description	
Dans le cas de la livraison d'un correctif ou d'une nouvelle version d'un composant applicatif, une procédure de migration doit être fournie par le fabricant.	
La procédure de migration a donné lieu à un ticket Redmine type "Requête technique" (SDETI) ou "Opération d'exploitation" (SDOP)	

2.1.3.2.3 Livrable/SNV2 - Paramétrage des applications SNV2 (variables Castor)

Étiquette du champ	Valeur du champ
ID de l'exigence	1070
Nom	Livrable/SNV2 - Paramétrage des applications SNV2 (variables Castor)
Etat exigence	23
Description	
Les paramètres de chaque livrable doivent pouvoir être exploités par le portail Technique. Voir documentation spécifique sur la configuration applicative sur le socle SX	

2.1.3.2.4 Livrable/SNV2 - Politiques Axway

Étiquette du champ	Valeur du champ
ID de l'exigence	1074
Nom	Livrable/SNV2 - Politiques Axway
Etat exigence	27
Description	
Les politiques Axway doivent être déployés pour chaque Endpoint mis à disposition sur les services SNV2	

2.1.3.2.5 Livrable/HAWAI - Nature du livrable

Étiquette du champ	Valeur du champ
ID de l'exigence	1069
Nom	Livrable/HAWAI - Nature du livrable

Etat exigence	23
Description	Les livrables sur plateforme HAWAI doivent être des RPM générés via l'outil FEDE contenant l'intégralité des composants applicatifs déployés sur une plateforme.

2.1.3.2.6 Livrable/HAWAI - Description des target

Étiquette du champ	Valeur du champ
ID de l'exigence	1075
Nom	Livrable/HAWAI - Description des target
Etat exigence	18
Description	La documentation du lot doit décrire les Targets Hawai à exécuter doivent être livrées. Les targets HAWAI permettent d'exécuter les tâches d'installation des composants sur une plateforme HAWAI

2.1.3.2.7 Livrable/HAWAI - Description base de données

Étiquette du champ	Valeur du champ
ID de l'exigence	1076
Nom	Livrable/HAWAI - Description base de données
Etat exigence	18
Description	Les informations suivantes concernant la base de données doivent être contenue dans la documentation du lot. Si update : Num version + target. Si la version de base de donnée ne change pas, NA doit être indiqué.

2.1.3.2.8 Livrable/HAWAI - Description des propriétés applicables

Étiquette du champ	Valeur du champ
ID de l'exigence	1077
Nom	Livrable/HAWAI - Description des propriétés applicables
Etat exigence	18
Description	La documentation du lot doit décrire les informations sur les propriétés HAWAI applicables. Si pas de nouvelles propriétés à considérer, NA doit être indiqué dans le doc du lot.

2.1.3.2.9 Livrable/HAWAI - Cohérence des versions

Étiquette du champ	Valeur du champ
ID de l'exigence	1078
Nom	Livrable/HAWAI - Cohérence des versions
Etat exigence	18
Description	

La concordance entre les livrables logiciels listés dans la doc de lot et les composants réellement livrés dans le dépôt SDIT doit être assurée.
Le numéro de version du livrable (rpm) ne doit pas dépasser 4 digits et ne doit contenir que des nombres (x.x.x.x max)

2.1.3.2.10 *Livrable/HAWAI - Localisation du stockage du DIT/DDP*

Étiquette du champ	Valeur du champ
ID de l'exigence	1079
Nom	Livrable/HAWAI - Localisation du stockage du DIT/DDP
Etat exigence	18
Description	La documentation du Lot doit préciser le lieu de stockage du DIT/DDP Note: Le lieu renseigné de stockage du DIT/DDP doit être valide, accessible et contient bien le DIT/DDP

2.1.3.2.11 *Livrable/HAWAI - Localisation DEX*

Étiquette du champ	Valeur du champ
ID de l'exigence	1080
Nom	Livrable/HAWAI - Localisation DEX
Etat exigence	18
Description	La documentation du Lot doit préciser le lieu de stockage du DEX Note: Le lieu renseigné de stockage du DEX doit être valide, accessible et contenir le DEX

2.1.3.2.12 *Livrable/HAWAI - Localisation specification fonctionnelle*

Étiquette du champ	Valeur du champ
ID de l'exigence	1081
Nom	Livrable/HAWAI - Localisation specification fonctionnelle
Etat exigence	18
Description	La documentation de lot doit contenir le lieu de stockage du dossier de spécification fonctionnelle

2.1.3.2.13 *Livrable/HAWAI - Politiques Axway*

Étiquette du champ	Valeur du champ
ID de l'exigence	1082
Nom	Livrable/HAWAI - Politiques Axway
Etat exigence	18
Description	Les politiques Axway doivent être déployés pour chaque service mis à disposition sur les backends HAWAI

2.1.3.2.14 Livrable/HAWAI - Document décrivant les composants

Étiquette du champ	Valeur du champ
ID de l'exigence	1083
Nom	Livrable/HAWAI - Document décrivant les composants
Etat exigence	18
Description	La documentation doit décrire la configuration des composants applicatifs sur la plateforme HAWAI

2.1.3.3 Livraison

Étiquette du champ	Valeur du champ
ID de l'exigence	1085
Nom	Livraison
Etat exigence	27
Description	

2.1.3.3.1 Livraison - Traçabilité d'une livraison (Redmine)

Étiquette du champ	Valeur du champ
ID de l'exigence	1086
Nom	Livraison - Traçabilité d'une livraison (Redmine)
Etat exigence	586
Description	La livraison d'un lot doit être symbolisée par un ticket redmine racine tracker Livraison rassemblant : • Les tickets redmines spécifiques aux différents socles • Les documents de livraison • Les différents livrables • La liste des dépôts modifiés • La liste des versions des artefacts produits et déployés (Charthelm, API, Agent) • L'URL des build de la chaîne de construction sur 'master' Si une livraison d'un livrable est refusée, un nouveau ticket Redmine doit être créé. Descriptif de l'attendu dans le ticket de livraison • Tracker : Livraison Projet : <nom_projet> Ex :P1, P6...etc • Statut : Livrée Fab • Assigné à : PRJ12_MOE_Ref_Tech • Description : <u>Livraison SNV2</u> - Lot SNV2 construit - Liste containers agent/api - Modification BD - Traitements batchs

Livraison HAWAI

- Version du RPM construit
- Frontend/backend/bd/agent
- Modification BD

Livraison PFS

- Version du chart livré
- Versions des images livrés
- Modification BD

Livraison Archimed

- Version des bundles livrés

Nouveau client

Référence documentaire

- * STD
- * Manuel Dev et Int
- * DEX
- * Rapport de tests Usine
- * Rapport qualité et sécurité

2.1.3.3.2 Livraison - Déversement des sources dans le dépôt URSSAF

Étiquette du champ	Valeur du champ
ID de l'exigence	1087
Nom	Livraison - Déversement des sources dans le dépôt URSSAF
Etat exigence	1257
Description	Toutes les sources des composants doivent être poussées dans les dépôts créés pour ce dernier : • Pour la livraison d'une version Release Candidate : dans la branche release/X.Y.Z • Pour la livraison d'une version X.Y.Z : dans la branche master Sur la branche Master, les sources de chaque composant doivent alors être taguées avec leur numéro de version sur 3 digits X.Y.Z (Release) X : version évolutive de production Y : version incrément de la version Z : version corrective de qualification SDIT/CNV Note: La branche master est à l'image de la version livrée à la conformité finale.

2.1.3.3.3 Livraison - Lancement des chaines de construction

Étiquette du champ	Valeur du champ
ID de l'exigence	1089
Nom	Livraison - Lancement des chaines de construction
Etat exigence	527
Description	

Suite au déversement des sources et à la création du tag, le fabricant doit lancer sur l'environnement URSSAF les différentes chaînes de construction menant au déploiement de l'application sur l'environnement cible.

2.1.3.3.4 Livraison/HAWAI - Traçabilité d'une livraison (Redmine)

Étiquette du champ	Valeur du champ
ID de l'exigence	1090
Nom	Livraison/HAWAI - Traçabilité d'une livraison (Redmine)
Etat exigence	20
Description	Chaque livrable produit par l'ESN titulaire émane de demandes d'évolutions ou de corrections d'anomalie effectuées par l'UCN. Pour les projets HAWAI, ces demandes sont consignées dans un le projet Redmine <à définir>. Ainsi chaque ticket Anomalie ou Evolution lié à une livraison doit répondre au modèle suivant : Projet : <A définir> Tracker : « Anomalie » ou « Evolution » Sujet : <Libellé du ticket> Version cible : <Version du composant > Statut : Livrée Fab (*) Assigné à : Raf – Cellule Intégration Echéance : <date de livraison> Description Descriptif de l'anomalie ou de l'évolution Demande(s) liée(s) à : Tracker Livraison type Livrable (voir exigence EX-RED02-v1)

2.1.3.3.5 Livraison/HAWAI - Mise à disposition du livrable (Redmine)

Étiquette du champ	Valeur du champ
ID de l'exigence	1091
Nom	Livraison/HAWAI - Mise à disposition du livrable (Redmine)
Etat exigence	18
Description	La mise à disposition du livrable Hawaii (RPM) doit être notifiée par un ticket de livraison. Un document de lot doit être attaché au ticket de livraison : #RED03

2.1.3.3.6 Livraison/HAWAI - Nomenclature du nom du lot

Étiquette du champ	Valeur du champ
ID de l'exigence	1092
Nom	Livraison/HAWAI - Nomenclature du nom du lot
Etat exigence	16
Description	Le nom de chaque lot doit suivre la nomenclature suivante:

Nomenclature du lot : YYMM00_CC_II

- YY = Année sur deux chiffres
- MM = Mois sur deux chiffres
- CC = Itérations CNV sur deux chiffres
- II = Itérations SDIT sur deux chiffres
- II = Itérations SDIT sur deux chiffres

2.1.3.3.7 Livraison/SNV2 - Déversement des Scripts migration

Étiquette du champ	Valeur du champ
ID de l'exigence	1095
Nom	Livraison/SNV2 - Déversement des Scripts migration
Etat exigence	25
Description	
La livraison d'une modification du schéma rénové dans le SNV2 impose obligatoirement la livraison des fichiers de configuration Liquibase® dans le dépôt Cobol CNTX. Le dépôt cible est le dépôt Cobol CNTX : http://gitlab.altair.recouv/319/cntx dans le répertoire « bdd ». La configuration du schéma « raf » est positionné à la racine de ce répertoire dans le fichier « raf.xml » Les évolutions du schéma sont positionnées dans des sous répertoires par version. Pour une évolution du schéma en v1.0, les fichiers de modification de base de données seront positionnés dans le sous répertoire « raf/v1.0 » Rn : Attention à la casse du numéro de version v[0-9].[0-9] Les fichiers doivent également être au format XML et l'encoding des fichiers doit être en UTF8.	

2.1.3.3.8 Livraison/SNV2 - Traçabilité de la livraison de Batches (Redmine)

Étiquette du champ	Valeur du champ
ID de l'exigence	1096
Nom	Livraison/SNV2 - Traçabilité de la livraison de Batches (Redmine)
Etat exigence	16
Description	
La livraison d'un batch est tracée dans Redmine via un ticket d'évolution ou d'anomalie dans le projet 319/CNTX avec les caractéristiques suivantes : Projet : SNV2 319/CNTX Tracker : « Evolution » Sujet : Traitement Batch <code_traitement V2> Version cible : <Version du lot V2 cible > Statut : Livrée Fab (*) Assigné à : Raf – Cellule Intégration Echéance : <date de livraison> Code traitement V2 : <code traitement V2> Code projet : <code projet refonte associé> Diagnostic Fonctionnel : Description Descriptif de l'anomalie ou de l'évolution	

Demande(s) liée(s) à :
• Tracker Livraison type Livrable

2.1.3.3.9 Livraison/SNV2 - Traçabilité de la livraison des containers (Redmine)

Étiquette du champ	Valeur du champ
ID de l'exigence	1458
Nom	Livraison/SNV2 - Traçabilité de la livraison des containers (Redmine)
Etat exigence	27
Description	La livraison des containers sur socle SX (base Oracle) est tracée dans Redmine via un ticket d'évolution dans le projet 319/CNTX avec les caractéristiques suivantes : Projet : SNV2 319/CNTX Tracker : « Evolution » Sujet : Container <nom container> Version <version schéma> Version cible : <Version du lot V2 cible > Statut : Livrée Fab (*) Assigné à : Raf – Cellule Intégration Echéance : <date de livraison> Code traitement V2 : <code traitement V2 container> Code projet : <code projet refonte associé> Diagnostic Fonctionnel : <Besoin> Intervention : <Solution> Description Descriptif de l'anomalie ou de l'évolution Demande(s) liée(s) à : • Tracker Livraison type Version (voir exigence EX-RED01-v1)

2.1.3.3.10 Livraison/SNV2 - Traçabilité de la livraison d'une Base de données (Redmine)

Étiquette du champ	Valeur du champ
ID de l'exigence	1097
Nom	Livraison/SNV2 - Traçabilité de la livraison d'une Base de données (Redmine)
Etat exigence	22
Description	La livraison de base de données du schéma RAF rénové sur socle SX (base Oracle) est tracée dans Redmine via un ticket d'évolution dans le projet 319/CNTX avec les caractéristiques suivantes : Projet : SNV2 319/CNTX Tracker : « Evolution » Sujet : Schéma RAF Rénové Version <version schéma> Version cible : <Version du lot V2 cible > Statut : Livrée Fab (*) Assigné à : Raf – Cellule Intégration Echéance : <date de livraison> Code traitement V2 : <code traitement V2 schéma RAF> Code projet : <code projet refonte associé>

Diagnostic Fonctionnel : <Besoin>
 Intervention : <Solution>
 Description
 Descriptif de l'anomalie ou de l'évolution

Demande(s) liée(s) à :
 • Tracker Livraison type Version (voir exigence EX-RED01-v1)

Exigence GIT associée : EX-GIT06

2.1.3.3.11 Livraison/PFS - Nomenclature du nom du lot

Étiquette du champ	Valeur du champ
ID de l'exigence	1098
Nom	Livraison/PFS - Nomenclature du nom du lot
Etat exigence	35
Description	Le nom de chaque lot doit suivre la nomenclature suivante: Nomenclature du lot : YYMM00_CC_II • YY = Année sur deux chiffres • MM = Mois sur deux chiffres • CC = Itérations CNV sur deux chiffres • II = Itérations SDIT sur deux chiffres

2.1.3.3.12 Livraison - Gestion des versions et tag git

Étiquette du champ	Valeur du champ
ID de l'exigence	1413
Nom	Livraison - Gestion des versions et tag git
Etat exigence	748
Description	Le versioning doit être géré au niveau des dépôts. Seuls les dépôts modifiés dans le cadre d'une livraison voient leur version évoluer.

2.1.3.3.13 Livraison/PFS - Gestion des versions

Étiquette du champ	Valeur du champ
ID de l'exigence	1414
Nom	Livraison/PFS - Gestion des versions
Etat exigence	39
Description	La version du ChartHelm (déclarée dans le Chart.yaml) définit la version globale de l'application. C'est la version qui est indiquée dans le ticket de livraison.

2.1.3.3.14 Livraison - Tracabilité des livraisons documentaires DocAsCode

Étiquette du champ	Valeur du champ
ID de l'exigence	1454
Nom	Livraison - Tracabilité des livraisons documentaires DocAsCode
Etat exigence	11
Description	Les livraison de documents de type doc as code doivent comporter les éléments suivants: • Dépot sur une branche du projet de documentation • La branche doit comprendre le nom du projet et increment suivi de "_livraison" (ex : p2i2_livraison) • Merge Request • Ticket redmine de type "Livraison documentaire" avec un lien vers la merge request • Aucun ajout de source ne doit être fait sur la branche une fois le ticket passé à "Livré par ESN"

2.1.3.3.15 Livraison - Déversement des sources - contrôles pre commits

Étiquette du champ	Valeur du champ
ID de l'exigence	2420
Nom	Livraison - Déversement des sources - contrôles pre commits
Etat exigence	6
Description	La qualité minimale du code doit être garantie à chaque commit. Les contrôles suivants doivent être inclus : * validité des fichiers yaml * validité des fichiers json * ajout d'une ligne en fin de fichier * lint sur les fichiers markdown Il est recommandé d'automatiser ces vérifications à l'aide de précommit hook. *Pré requis:* * Installation de python 3 sur poste de dev * Installation de modules python : ** 'python -m pip install pre-commit' (ou 'pip install pre-commit' pour linux): https://pre-commit.com/ *Mise en oeuvre:* * Dans chaque projets/modules mettre en place le pre-commit en ajoutant le fichier '.pre-commit-config.yaml' à la racine du projet. * Modification du manuel Dev/Int pour détailler l'installation du pre-commit sur un nouveau poste de développement

2.1.3.4 Déploiement

Étiquette du champ	Valeur du champ
ID de l'exigence	1099

Nom	Déploiement
Etat exigence	21
Description	

2.1.3.4.1 Point de contrôle/SNV2 - Contrôle CTRL-1-Path-Image-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1100
Nom	Point de contrôle/SNV2 - Contrôle CTRL-1-Path-Image-1
Etat exigence	26
Description	
	Dans le fichier etcv2/ws-java-deploy, le chemin de l'image livrée doit toujours contenir registry:5000
	Regexp sur le contenu de la propriété Image <code>"^registry:5000/acoss/recouv(.)+"</code>

2.1.3.4.2 Point de contrôle/SNV2 - Contrôle CTRL-2-Path-Image-2

Étiquette du champ	Valeur du champ
ID de l'exigence	1101
Nom	Point de contrôle/SNV2 - Contrôle CTRL-2-Path-Image-2
Etat exigence	21
Description	
	Dans le fichier etcv2/ws-java-deploy, le chemin de l'image livrée doit suivre la nomenclature suivante : /acoss/recouv/<fonction>/<libre> avec <fonction>ga, gce, gci, raf, ctrl, demo, trans
	Regexp sur le contenu de la propriété Image <code>"^(.)+/(.)+/(.)+/(ga gce gci raf ctrl demo trans)/(.)+"</code>

2.1.3.4.3 Point de contrôle/SNV2 - Contrôle CTRL-3-Nom-Image -1

Étiquette du champ	Valeur du champ
ID de l'exigence	1102
Nom	Point de contrôle/SNV2 - Contrôle CTRL-3-Nom-Image -1
Etat exigence	21
Description	
	Dans le fichier etcv2/ws-java-deploy, le nom de l'image livrée doit suivre la nomenclature suivante : <type>-<fonction>-<fonctionnalité> <type> : "agt" ou "api" pour agent (daemon) ou webapp exposant une api <fonction> : ga, gce, gci, raf, ctrl, demo, trans <fonctionnalité> : libellé libre pour donner du sens au nom
	Regexp sur le contenu de la propriété Image <code>"^(.)+/(.)+/(.)+/(agt api ws)-(ga gce gci raf jobde ctrl demo trans)-(.)+"</code>

2.1.3.4.4 Point de contrôle/SNV2 - Contrôle CTRL-4-Version-Image-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1103
Nom	Point de contrôle/SNV2 - Contrôle CTRL-4-Version-Image-1
Etat exigence	22
Description	Dans le fichier etcv2/ws-java-deploy, la version de l'image livrée doit suivre la nomenclature suivante : <version_image> doit contenir 4 digits <MAJEURE>.<MINEURE>.<CORRECTIF>-<ITDEV> <MAJEURE> : (début 1) <MINEURE> : (début 0) <CORRECTIF> : (début 0) <ITDEV> : (début 0) Regex sur le contenu de la propriété Image "^(.)+/(.)+/(.)+/(.)+-(.)+-(.)+:[1-9]{1}.[0-9]{1}.[0-9]{1}-[0-9]{1}"

2.1.3.4.5 Point de contrôle/SNV2 - Contrôle CTRL-5-Nom-Image -2

Étiquette du champ	Valeur du champ
ID de l'exigence	1104
Nom	Point de contrôle/SNV2 - Contrôle CTRL-5-Nom-Image -2
Etat exigence	21
Description	Dans le fichier etcv2/ws-java-deploy, le nom de l'image ne doit pas dépasser 18 caractères
	Par programmation, if length (nom-image) > 18 alors contrôle KO

2.1.3.4.6 Point de contrôle/SNV2 - Contrôle CTRL-6-Nom-Image-3

Étiquette du champ	Valeur du champ
ID de l'exigence	1105
Nom	Point de contrôle/SNV2 - Contrôle CTRL-6-Nom-Image-3
Etat exigence	21
Description	Dans le fichier etcv2/ws-java-deploy, lors du contrôle cohérence entre les propriétés « name » et « image » les champs <nom-image> dans la propriété « name » et « image » doivent être identiques. Par programmation Extraction contenu propriété de <name> extraction nom-image de la propriété <image> Si les 2 sont différents alors contrôle KO

2.1.3.4.7 Point de contrôle/SNV2 - Contrôle CTRL-7-Name-Type-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1106
Nom	Point de contrôle/SNV2 - Contrôle CTRL-7-Name-Type-1
Etat exigence	21
Description	
Si le nom du conteneur se termine par « -p » alors le type du conteneur doit être égal à « p »	
Par programmation, Extraction des 2 derniers caractères de la propriété Name Si Name se termine par '-p' et Type <> "p" Alors contrôle KO	
Si le nom du conteneur se termine par « -c » alors le type du conteneur doit être égal à « c »	
Par programmation, Extraction des 2 derniers caractères de la propriété Name Si Name se termine par '-c' et Type <> "c" Alors contrôle KO	

2.1.3.4.8 Point de contrôle/SNV2 - Contrôle CTRL-8-Erb-Nom

Étiquette du champ	Valeur du champ
ID de l'exigence	1107
Nom	Point de contrôle/SNV2 - Contrôle CTRL-8-Erb-Nom
Etat exigence	21
Description	
La nomenclature du fichier doit être la suivante : <nom_container>.erb	
Par programmation; extraction de tous les noms des images fichier ws-java-deploy.json, et vérification présence d'un fichier .erb de même nom	

2.1.3.4.9 Point de contrôle/SNV2 - Contrôle CTRL-9-Erb-Encoding

Étiquette du champ	Valeur du champ
ID de l'exigence	1108
Nom	Point de contrôle/SNV2 - Contrôle CTRL-9-Erb-Encoding
Etat exigence	23
Description	
L'encoding du fichier doit être en UTF-8	
Par programmation exécution de la commande "File" sur chaque fichier .erb, son type doit être = UTF-8 Unicode text	

2.1.3.4.10 Point de contrôle/SNV2 - Contrôle CTRL-10-Nom-Image-4

Étiquette du champ	Valeur du champ
ID de l'exigence	1395
Nom	Point de contrôle/SNV2 - Contrôle CTRL-10-Nom-Image-4
Etat exigence	27
Description	L'encoding du fichier doit être en UTF-8 Par programmation exécution de la commande "File" sur chaque fichier .erb, son type doit être = UTF-8 Unicode text

2.1.3.4.11 Point de contrôle/SNV2 - Contrôle CTRL-11-NomService-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1109
Nom	Point de contrôle/SNV2 - Contrôle CTRL-11-NomService-1
Etat exigence	22
Description	Le nom du service doit vérifier l'expression régulière suivante : "[0-9A-Za-z_-]+\$" regexp: "[0-9A-Za-z_-]+\$"

2.1.3.4.12 Point de contrôle/SNV2 - Contrôle CTRL-12-TypeVM-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1110
Nom	Point de contrôle/SNV2 - Contrôle CTRL-12-TypeVM-1
Etat exigence	22
Description	Propriété : Type de VM Les valeurs acceptées pour le Type de VM sont c ou p Par programmation

2.1.3.4.13 Point de contrôle/SNV2 - Contrôle CTRL-13-ImageProd-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1111
Nom	Point de contrôle/SNV2 - Contrôle CTRL-13-ImageProd-1
Etat exigence	22
Description	L'image docker doit être présente sur un des registries de production Par programmation exécution de la commande podman pull

2.1.3.4.14 Point de contrôle/SNV2 - Contrôle CTRL-14-OnlyOn-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1112
Nom	Point de contrôle/SNV2 - Contrôle CTRL-14-OnlyOn-1
Etat exigence	22
Description	La propriété only_on contient une liste d'URCODE séparés par des virgules. L'automate fait une vérification syntaxique de la déclaration des URCODE. Regexp: "[0-9]{3}\$"

2.1.3.4.15 Point de contrôle/SNV2 - Contrôle CTRL-15-Bascule-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1113
Nom	Point de contrôle/SNV2 - Contrôle CTRL-15-Bascule-1
Etat exigence	22
Description	Le propriété bascule_S n'est autorisé que pour les VM de type « p » Par programmation

2.1.3.4.16 Point de contrôle/SNV2 - Contrôle CTRL-16-PlagePort-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1114
Nom	Point de contrôle/SNV2 - Contrôle CTRL-16-PlagePort-1
Etat exigence	22
Description	Le mappage sur plusieurs ports n'est pas autorisé.

2.1.3.4.17 Point de contrôle/SNV2 - Contrôle CTRL-17-ImageDev-1

Étiquette du champ	Valeur du champ
ID de l'exigence	1115
Nom	Point de contrôle/SNV2 - Contrôle CTRL-17-ImageDev-1
Etat exigence	22
Description	L'image docker doit être présente sur la registries de dev Par programmation exécution de la commande podman pull

2.1.3.4.18 Point de contrôle/PFS - Contrôle nommage

Étiquette du champ	Valeur du champ
ID de l'exigence	1116
Nom	Point de contrôle/PFS - Contrôle nommage
Etat exigence	25
Description	La norme pour le nommage des tags devait être le numéro de l'itération livrée. Par exemple : - itération 220500_00_00 --> tag 220500_00_00 - itération 220500_02_01 --> tag 220500_02_01 - itération 220610_00_00 --> tag 220610_00_00

2.1.4 Contraintes d'observabilité

Étiquette du champ	Valeur du champ
ID de l'exigence	1533
Nom	Contraintes d'observabilité
Etat exigence	4
Description	

2.1.5 Contraintes de Performance

Étiquette du champ	Valeur du champ
ID de l'exigence	1612
Nom	Contraintes de Performance
Etat exigence	14
Description	

2.1.5.1 Performance - Consommateur de messages

Étiquette du champ	Valeur du champ
ID de l'exigence	3231
Nom	Performance - Consommateur de messages
Etat exigence	9
Description	Le temps de consommation d'un message (I.E. entre le moment où il est pris en compte et le moment où il est acquitté au niveau du broker) doit être en production : • ≤ 100 ms s'il s'agit d'un message à traiter • ≤ 50 ms s'il s'agit d'un message à écarter

2.1.5.2 Performance - Webservice

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	3232
Nom	Performance - Webservice
Etat exigence	10
Description	Le système doit être capable de soutenir un débit minimal de 200 transactions par seconde (TPS) tous endpoints confondus , tout en garantissant un temps de réponse moyen ≤ 500 ms et un 95 ^e percentile ≤ 700 ms, sur un environnement de PROD

2.1.5.3 Performance - IHM

Étiquette du champ	Valeur du champ
ID de l'exigence	3233
Nom	Performance - IHM
Etat exigence	8
Description	Pour 100 appels simultanés sur une même page, le temps de réponse (chargement complet) de cette page doit être ≤ 3 secondes (Temps global de réalisation de l'ensemble des appels http nécessaire à ce qu'elle soit affichée et disponible pour l'utilisateur) Sur la base d'environnement ISO PROD

2.1.5.4 Performance - Dépassement de limites

Étiquette du champ	Valeur du champ
ID de l'exigence	3235
Nom	Performance - Dépassement de limites
Etat exigence	8
Description	Dans le cas d'une sollicitation supérieure aux limites des exigences, le système doit avoir la capacité d'absorber cette sollicitation sans impact ou dégradation du contexte interne ou externe du composant même si la performance est non conforme aux attendus des exigences. (Non applicable aux consommateurs de messages pour lesquels le broker accumule les sollicitations)

2.1.6 ENF Dépréciées ou Annulées

Étiquette du champ	Valeur du champ
ID de l'exigence	2553
Nom	ENF Dépréciées ou Annulées
Etat exigence	10
Description	

2.1.6.1 Backend/PFS - Communication inter-namespace

Étiquette du champ	Valeur du champ
ID de l'exigence	866
Nom	Backend/PFS - Communication inter-namespace
Etat exigence	42
Description	<i>La communication entre services appartenant à des namespace PFS différents doit se faire en passant par Archimed avec le protocole HTTPS</i> Dorénavant couvert par : ENF-PFSV2011 Médiation Archimed des API Web

2.1.6.2 Backend/PFS - Communication intra-namespace

Étiquette du champ	Valeur du champ
ID de l'exigence	867
Nom	Backend/PFS - Communication intra-namespace
Etat exigence	37
Description	<i>La communication entre services appartenant à un même namespace se fait en utilisant les services k8s pod-service-pod</i> Dorénavant couvert par : ENF-PFSV2011 Médiation Archimed des API Web

2.1.6.3 Performance

Étiquette du champ	Valeur du champ
ID de l'exigence	3236
Nom	Performance
Etat exigence	7
Description	

2.1.6.3.1 A - IHM

Étiquette du champ	Valeur du champ
ID de l'exigence	1613
Nom	A - IHM
Etat exigence	9
Description	

2.1.6.3.2 B - Webservices/API

Étiquette du champ	Valeur du champ
--------------------	-----------------

ID de l'exigence	1614
Nom	B - Webservices/API
Etat exigence	11
Description	

2.1.6.3.3 C - Batch

Étiquette du champ	Valeur du champ
ID de l'exigence	1615
Nom	C - Batch
Etat exigence	16
Description	

2.1.6.3.3.1 RAF-PRF-Traitement de reprise inférieur à 24h

Étiquette du champ	Valeur du champ
ID de l'exigence	1616
Nom	RAF-PRF-Traitement de reprise inférieur à 24h
Etat exigence	24
Description	Objectif de l'exigence : Le traitement de reprise RAF doit prendre moins de 24h Cas d'utilisation : Fonctionnalité associée : Historique des trains :

2.1.6.3.3.2 RAF-PRF-Taux de traitement pour synchronisation des écarts négatifs = 150 000/h

Étiquette du champ	Valeur du champ
ID de l'exigence	1617
Nom	RAF-PRF-Taux de traitement pour synchronisation des écarts négatifs = 150 000/h
Etat exigence	20
Description	Objectif de l'exigence : Le traitement de synchronisation des écarts négatifs doit afficher un taux de traitement de 150 000 écarts négatifs/h Cas d'utilisation : Fonctionnalité associée : Historique des trains :

2.1.6.3.3.3 RAF-PRF-Contrôle de la taille des FS exfv2 et bddv2 avant et après le traitement de reprise des données

Étiquette du champ	Valeur du champ
ID de l'exigence	1618
Nom	RAF-PRF-Contrôle de la taille des FS exfv2 et bddv2 avant et après le traitement de reprise des données
Etat exigence	24
Description	Objectif de l'exigence : Il faut contrôler l'évolution des FS exfv2 et bddv2 avant et après le traitement de reprise des données Cas d'utilisation : Fonctionnalité associée : Historique des trains :