

# Standards SI tl

## Table des matières

<b><u>1</u></b>	<b><u>Glossaire</u></b> .....	<b><u>2</u></b>
<b><u>2</u></b>	<b><u>Information sur le document</u></b> .....	<b><u>2</u></b>
2.1	Revue et approbation .....	2
2.2	Historique du document .....	2
<b><u>3</u></b>	<b><u>Introduction</u></b> .....	<b><u>3</u></b>
3.1	Au sujet de ce document .....	3
3.2	Audience .....	3
<b><u>4</u></b>	<b><u>Gestion des versions et obsolescence</u></b> .....	<b><u>3</u></b>
<b><u>5</u></b>	<b><u>Environnements</u></b> .....	<b><u>4</u></b>
<b><u>6</u></b>	<b><u>Architecture cible et Intégration</u></b> .....	<b><u>5</u></b>
6.1	Principes généraux .....	5
6.2	Embarqué .....	6
6.3	Architecture applicative globale .....	8
<b><u>7</u></b>	<b><u>Protection des données</u></b> .....	<b><u>9</u></b>
<b><u>8</u></b>	<b><u>Sécurité – EN COURS DE REVISION</u></b> .....	<b><u>10</u></b>
8.1	Sécurité et protection des données .....	10
8.2	Sécurité du réseau .....	10
8.3	Sécurité des applications .....	10
8.4	VPN .....	10
<b><u>9</u></b>	<b><u>Informatique utilisateurs</u></b> .....	<b><u>11</u></b>
9.1	Matériel .....	11
9.2	Systèmes d'exploitation .....	11
9.3	Logiciels .....	11
<b><u>10</u></b>	<b><u>Informatique mutualisée</u></b> .....	<b><u>12</u></b>
10.1	Matériel et infrastructure .....	12
10.2	Systèmes d'exploitation, virtualisation et container .....	13
10.3	Logiciels .....	14
10.4	Convention de nommage .....	15
<b><u>11</u></b>	<b><u>Environnement de développement et d'intégration – Gestion de projet</u></b> .....	<b><u>16</u></b>

## 1 Glossaire

<a href="#">ESB</a>	Enterprise Service Bus ()
<a href="#">LB</a>	Load Balancer
<a href="#">MQTT</a>	Message Queuing Telemetry Transport
<a href="#">REST</a>	REpresentational State Transfer
<a href="#">RP</a>	Reverse Proxy
<a href="#">UI</a>	User Interface

## 2 Information sur le document

### 2.1 Revue et approbation

	Nom	Rôle
Responsable du document	Giorgio Anastopoulos	Architecture SI
Validation	Cédric Lab	Responsable SI
Revue	David Mayor	Responsable ISP
Revue	Stéphane Beaussault	Responsable DIS

### 2.2 Historique du document

Version	Auteur	Date	Commentaires/Changement
1.0	Jean-Philippe Suter		Version initiale
1.1	David Mayor	01.07.2016	Mise à jour semestrielle
1.2	Giorgio Anastopoulos	03.12.2018	Révision majeure avec la collaboration des responsables d'équipe et de tous les collègues.
1.3	Giorgio Anastopoulos	01.07.2019	Révision mineure semestrielle.
1.4	Giorgio Anastopoulos	06.01.2020	Export de données d'une solution. Compléments sur la sécurité réseau. Standards de nommage API, messaging et serveur. Ajout de TypeScript + NoteJS pour les technologies backend.

### 3 **Introduction**

#### 3.1 **Au sujet de ce document**

Ce document décrit la liste des logiciels, matériels et méthodes de travail standards pour le développement, le déploiement et la gestion de matériels, logiciels et solutions informatiques aux tl. Il décrit également les exigences d'architecture et sécurité auxquelles les solutions logicielles doivent répondre afin de s'intégrer de manière harmonieuse dans le SI des tl. Pour terminer, il définit les différents types d'environnement possibles dans le SI des tl.

Ce document est annexé aux appels d'offre comportant au moins une composante IT et définit des exigences techniques sur lesquelles les solutions proposées seront évaluées. Il ne définit aucune règle ou critère éliminatoire et chaque exigence ne sera pondérée et évaluée dans un appel d'offre que si elle est pertinente. Le caractère éliminatoire d'une règle est déterminé au cas par cas en fonction de l'appel d'offre.

Ce document représente une vue à un instant donné et peut changer à tout moment afin d'être adapté aux évolutions technologiques et organisationnelles. La version faisant foi dans le cadre d'un appel d'offres est celle en vigueur à l'instant de la publication.

#### 3.2 **Audience**

Ce document est destiné à toute l'équipe SI ainsi qu'à toute personne collaborant avec SI, que ce soit dans le cadre d'un projet, d'un incident, d'une demande ou d'un appel d'offre.

### 4 **Gestion des versions et obsolescence**

Ce document n'indique de manière générale pas de numéro de version ou d'autres informations détaillées sur les logiciels ou matériels, ceci car ces informations peuvent évoluer rapidement et ainsi rendre le document obsolète.

Une nouvelle version de solution avec évolutions retro-compatibles implique que toutes les fonctionnalités de la version précédente sont présentes dans la nouvelle version et se comportent de la même manière (hors correction de bug). Une telle version n'implique donc aucune évolution pour le reste du SI, en particulier au niveau des échanges.

Une nouvelle version majeure indique une évolution importante de la solution. En principe elle comporte des évolutions non retro-compatibles, qui doivent être clairement documentées et communiquées afin de prévoir les éventuelles modifications nécessaires au niveau du SI, en particulier au niveau des échanges de données.

Numérotation de version proposée par tl :

- Sur trois nombres, séparés par des points → X.Y.Z
- Un incrément de X implique une version majeure pour refléter une évolution d'architecture ou fonctionnelle importante, par exemple nouvelle interface ou nouveau module, qui peut être rétro-incompatible. Reflété dans une roadmap stratégie.
- Un incrément de Y implique une version mineure pour refléter une évolution d'architecture ou fonctionnelle avec rétrocompatibilité. Reflété dans une roadmap opérationnelle.
- Z : numéro de patch, change lorsqu'il y a des corrections apportées à des fonctionnalités déjà déployées, sans évolution d'architecture ou fonctionnelle
- En cas de perte de rétrocompatibilité, les fonctionnalités qui sont touchées sont clairement documentées, ainsi que le processus de migration

Les règles suivantes sont appliquées afin d'éviter l'obsolescence logicielle et matérielle :

Référence	Description
-----------	-------------

<b>R-VERS-01</b>	Dans aucun cas le matériel ou logiciel est utilisé au-delà de la date limite de support. Toute solution n'utilise que des composants qui sont supportés par leur éditeur ou en garanti le support.
<b>R-VERS-02</b>	De manière générale, la mise à jour d'un logiciel ou matériel est planifiée dès deux versions majeures de retard sur la dernière version majeure disponible, mais au plus tard suffisamment de temps à l'avance pour que sa réalisation puisse se terminer avant la fin de support de la version utilisée.
<b>R-VERS-03</b>	Tout contrat de maintenance d'une solution achetée inclut la mise à jour de la solution par rapport à son évolution et à celle des versions de l'un ou l'autre des composants qu'elle utilise. Par exemple, la mise à disposition d'une nouvelle version de solution logicielle doit être effective avant la fin de support de l'un ou l'autre des composants qu'elle utilise.
<b>R-VERS-04</b>	La sémantique du numéro de version est définie. En particulier, en fonction du changement du numéro de version il est possible de distinguer une évolution majeure, une évolution non retro-compatibles ou une évolution retro-compatible.

## 5

### Environnements

Le tableau ci-dessous définit les différents types d'environnement qui peuvent exister aux tl, ainsi que leur fonction.

Nom	Objectifs	Nb	Mises à jour	Utilisateurs
Bac à Sable (SBX)	Tester un concept, une nouvelle technologie, un upgrade	N	From scratch Très Fréquentes	Développeurs
Développement (DEV)	Développement Tests unitaires	N	Stable 1 à 2 fois par an	Développeurs
Intégration (INT)	Tests d'intégration (Vérification des interfaces et de l'architecture)	N	Stable Chaque intégration	Développeurs Intégrateurs
Qualité (QAS)	Test système (Validation des exigences fonctionnelles)	1	Très stable Chaque livraison	Testeurs QA Key Users
Formation (TRN)	Formation des utilisateurs finaux	1	Très stable Chaque formation	Formateurs Utilisateurs finaux Administrateurs
Production (PRD)	Tests de recette (Validation de toutes les exigences dans les conditions réelles)	1	Très stable Chaque livraison	Utilisateurs finaux dont Key Users

## 6 Architecture cible et Intégration

### 6.1 Principes généraux

L'architecture cible du système d'information des tl est illustrée par le schéma de la Figure 1 et se construit sur la base des principes suivants :

- L'ensemble des données de la solution doit pouvoir être exporté dans un format qui est décrit en détail afin de pouvoir être lu et utilisé par un autre système.
- Si les applications fournissent ou consomment des API REST et qu'aucune conversion de format n'est nécessaire, alors elles communiquent directement par appels d'API REST (cf. APP1, APP2)
- Les appels d'API REST se font à travers le RP/LB, ce qui permet, entre autres, de gérer la sécurisation du canal de communication (HTTPs), la récolte de statistiques, l'analyse de problèmes, ...
- Si l'application ne peut pas communiquer par REST ou qu'une conversion de format est nécessaire, alors l'information qu'elle fournit ou nécessite est traitée ou mise à disposition par le biais de l'ESB, cela à chaque fois qu'il s'avère nécessaire d'échanger avec d'autres applications métier ou de monitoring ; l'ESB se positionne dans ce cas comme intermédiaire facilitateur pour l'implémentation des échanges inter-applicatifs (cf. APP3, APP4, APP5) et permet de traiter tout protocole standard (SFTP, FTP(s), HTTP(s), LDAP, ...) et format de fichier basé sur du texte (CSV, JSON, XML, positionnel, ...)
- Plusieurs applications/composants/modules d'une même solution peuvent communiquer entre eux en utilisant un protocole propriétaire (cf. APP3, APP4) ; néanmoins et à chaque fois que cela est requis par les besoins ou contraintes du SI de tl, ils publient ces informations de manière à ce qu'elles soient exploitables par d'autres systèmes, ceci selon les modalités décrites dans ce même chapitre.

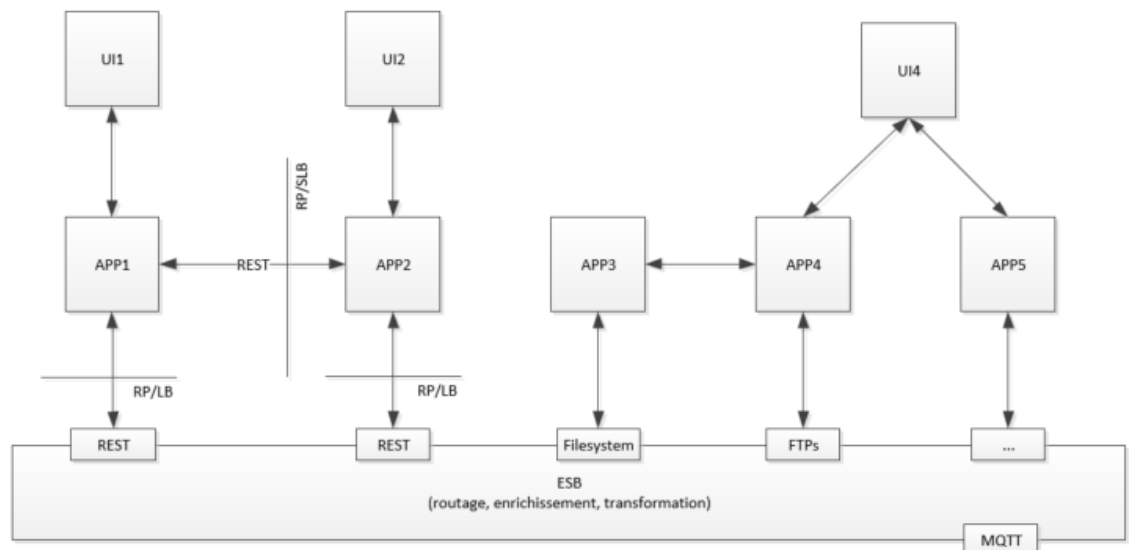


Figure 1 Architecture cible du SI des tl

Comme l'illustre le schéma ci-dessus, les applications peuvent communiquer directement par API REST (APP1, APP2), par protocole/format propriétaire ou tout autre protocole/format (APP3, APP4). Dans tous les cas, les applications mettent (également) à disposition ou consomment des données d'autres applications pour le bon fonctionnement du SI des tl. Ces échanges se font sur la base de protocoles et formats standards et ouverts.

L'informatique des tl dispose d'un outil de type ESB pour faciliter la mise en place des échanges.

Les solutions respectent les règles et exigences suivantes :

Référence	Description
R-ARCH-01	L'intégration se fait par les services et pas par les données. En particulier, aucun système n'accède directement les données dans la base de données d'un autre. L'échange de fichiers est toléré mais non encouragé.
R-ARCH-02	La solution met à disposition les données nécessaires au bon fonctionnement du SI dans un format décrit et utilisable, sans contrainte technique/technologique, de préférence dans un format textuel ou dans tous les cas dans un format facilement utilisable.
R-ARCH-03	La solution est capable de consommer des API REST afin d'obtenir ou transmettre des informations de/à d'autres applications.
R-ARCH-04	La solution fournit des API REST pour la mise à disposition d'informations à d'autres applications ou pour que d'autres applications puissent lui transmettre des données.
R-ARCH-05	Les API REST respectent les standards usuels et, pour les parties non définies (ex. pagination, ...), les standards d'un grand acteur du web (par exemple Google).
R-ARCH-06	Les possibilités d'échange de données, par API REST ou autre (filesystem, DB, FTPs, ...), sont décrites et accompagnées d'exemples qui en éclairent le fonctionnement.
R-ARCH-07	Tout échange entre deux applications non intégrées s'effectue par le biais d'un ESB.
R-ARCH-08	L'ensemble des données de la solution doit pouvoir être exporté dans un format qui est décrit en détail et peut être lu et utilisé par un autre système.

## 6.2 Embarqué

Les logiciels embarqués fonctionnent sur un PC de type :

- Core i3 1.7GHz
- 4Gb Ram
- SSD 8Gb
- OS Base Linux Yocto-OpenEmbedded

Par ordre de préférence, seront privilégiées les solutions qui :

- Ne dépendent pas du système d'exploitation ou ne sont pas liées à un unique système d'exploitation
- S'exécutent sur l'OS Base Linux Yocto-OpenEmbedded

- S'exécutent sur l'OS Windows

Une solution livrée sous forme de runtime Docker sera privilégiée, afin de partager les ressources et assurer la bonne coexistence des différents logiciels.

Hors solution Docker, le partenaire fournit un logiciel embarqué qui doit fonctionner en utilisant des composants, bibliothèques standards du système d'exploitation de l'embarqué défini par tl.

Les solutions de l'embarqué sont certifiées compatibles avec norme [ITxPT](#).

L'architecture d'intégration cible des applications embarquées du système d'information des tl est illustrée par le schéma de la Figure 2.

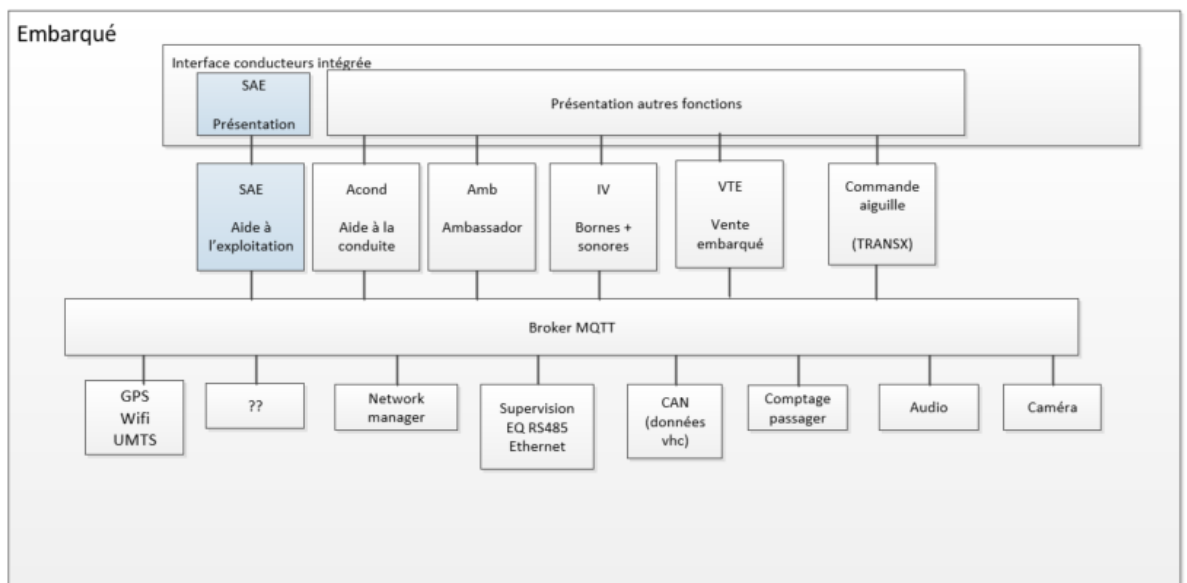


Figure 2 Architecture cible des systèmes embarqués du SI des tl

Le principe fondamental est que les systèmes communiquent par le biais d'un broker MQTT en y publiant ou consommant des informations de manière compatible avec la norme EN13149. À défaut, la préférence est pour des communications REST.

Le broker MQTT de l'embarqué est connecté à l'ESB présenté dans la Figure 1 et toute information publiée dans l'un peut être transférée vers l'autre pour mise à disposition des solutions qui y sont connectées.

Les solutions respectent les règles suivantes en plus de celles décrites dans 6.1:

Référence	Description
R-ARCH-08	La solution fonctionne sur un PC tel que décrit ci-dessus.
R-ARCH-09	La solution ne dépend pas du système d'exploitation. Si ce n'est pas le cas elle s'exécute sur l'OS Base Linux Yocto-OpenEMbedded. Si ce n'est pas le cas elle s'exécute sur un OS Linux ou Windows.
R-ARCH-10	La solution est livrée sous forme de Runtime Docker.
R-ARCH-11	La solution sait communiquer avec un broker MQTT pour consommer les données publiées par d'autres systèmes et qui lui sont nécessaires ou pour y publier les données nécessaires au bon fonctionnement d'autres systèmes.

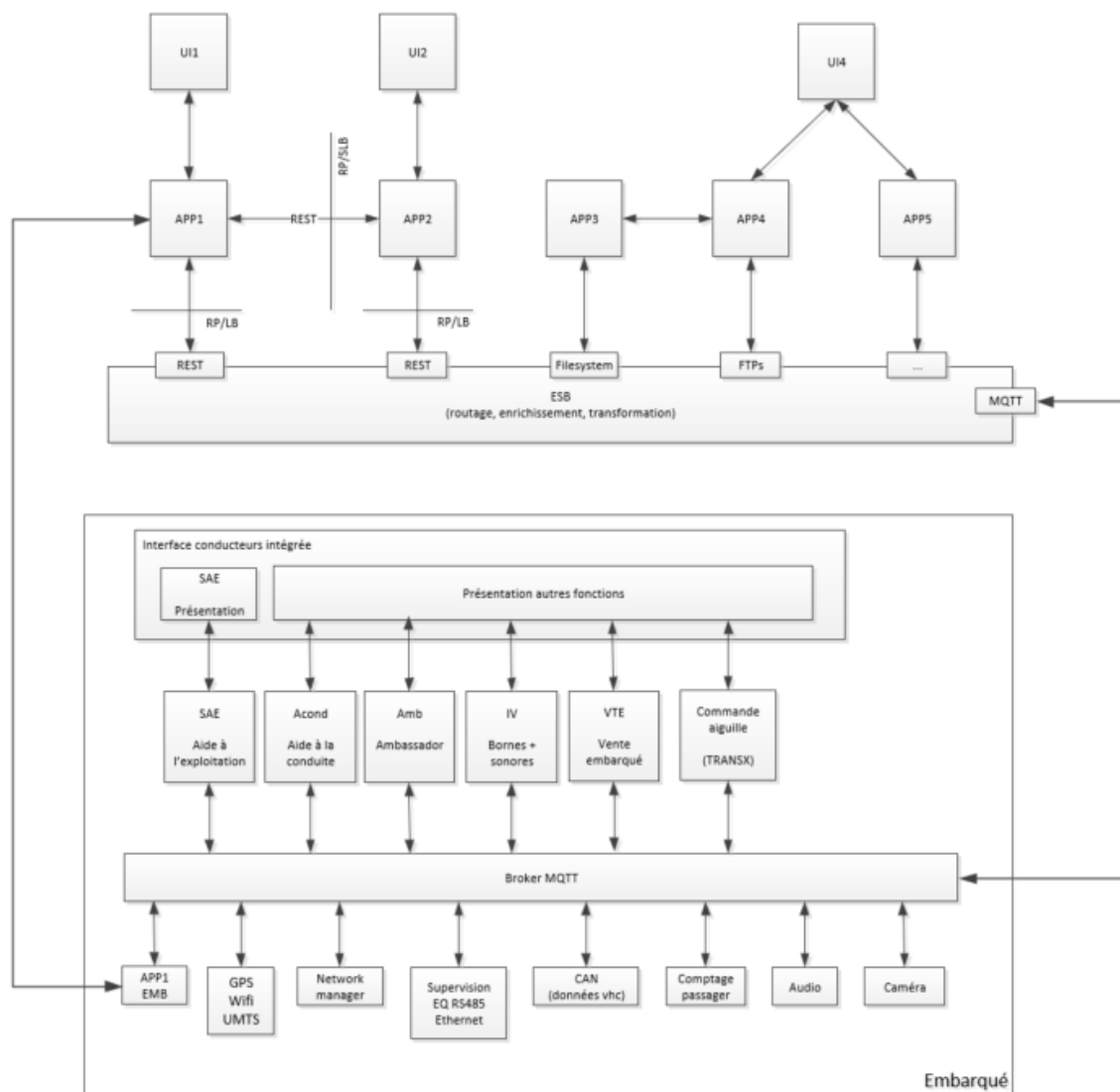
R-ARCH-12	À défaut, les protocoles et formats supportés par la solution sont facilement utilisables et décrits de manière détaillée, permettant une implémentation efficace.
R-ARCH-13	La solution est certifiée compatible avec la norme <a href="#">ITxPT</a> .

### 6.3 Architecture applicative globale

La Figure 3 illustre l'architecture applicative globale, obtenue par superposition de du schéma de la Figure 1 et de la Figure 2.

Il est à remarquer que le broker du système embarqué est connecté à l'ESB par le biais d'un connecteur mis en place dans ce dernier. C'est par ce biais que les informations publiées dans le broker de l'embarqué remontent et sont mises à disposition dans les applications backend. Le broker MQTT prend en charge les messages dans l'embarqué et assure leur livraison même en cas de plantage du PC, il est donc à privilégier comme moyen de communication avec le backend car il assure la transmission des informations même en cas de coupure de réseau en les transférant dès que ce dernier redevient disponible.





*Figure 3 Architecture applicative globale*

Pour une même solution, les différents composants backend et déployés sur l'embarqué peuvent communiquer directement par un protocole à choix (ex APP1 EMB et APP1). Ceci est également le cas de composants de solutions différentes qui s'intègrent par nature.

Dans tous les cas, ils doivent mettre à disposition les informations nécessaires au bon fonctionnement des autres applications du SI des tl en utilisant un format et un protocole qui sont documentés de manière détaillée.

## 7 Protection des données

Le traitement des données doit respecter la Loi sur la Protection des Données (LPD) et le Règlement Général à la Protection des Données (RGPD).

## 8 **Sécurité – EN COURS DE REVISION**

Toute communication vers une UI ou application hors du réseau tl doit être encryptée. Le choix est laissé pour les communications au sein de notre LAN, la préférence étant donnée à l'encryption lorsque cela est simple.

L'authentification se base sur le standard SAML2 à chaque fois que cela est possible.

Les applications web respectent le standard [OWASP](#).

Les applications mettent à disposition des points de monitoring (API REST ou autre) qui puissent être utilisés par un outil ad hoc pour en surveiller la disponibilité.

### 8.1 **Sécurité et protection des données**

### 8.2 **Sécurité du réseau**

Le réseau tl est constitué d'une DMZ et plusieurs sous-réseaux internes ainsi que des réseaux fermés sécurisés pour des systèmes comme SCADA, GTC tunnel, etc. Les sous-réseaux sont séparés par des firewalls NEXTGEN

L'accès au réseau tl depuis internet se fait par le biais de VPN (cf. §8.4).

IPV6 n'est pas géré.

L'encryptage se fait par le biais de clés de 256 bits au minimum.

La sécurité du réseau wifi se fait par le biais de WPA2 + 802.1X

### 8.3 **Sécurité des applications**

La sécurité backend est implémentée indépendamment de celle mise en place dans le frontend. Il faut partir du principe que le backend peut être accédé par d'autres sources que son frontend.

Pour le cas des applications exposées sur Internet, le frontend est déployé en DMZ et le backend est déployé dans un réseau interne.

### 8.4 **VPN**

En règle générale, aucun accès VPN n'est donné sur les ressources de production. Les besoins spécifiques sont étudiés au cas par cas et une solution alternative sera systématiquement privilégiée.

Lorsque tl est amené à le faire, l'accès VPN est accordé pour les applications d'infogérance. Dans ce contexte, deux cas se distinguent :

- Nominatif : l'accès est donné à un particulier d'une entreprise qui est connu par son nom. Cela s'effectue par le biais de client lourd VPN.
- Entreprise : l'accès est donné à une adresse IP publique d'entreprise sur un pool de ressources via un port et un protocole. Cela s'effectue par tunnel IPSEC.

Les solutions respectent les règles suivantes :

Référence	Description
R-SEC-01	Les échanges de/vers des systèmes hors du réseau tl sont encryptés.
R-SEC-02	La solution supporte les communications HTTP encryptées (HTTPS).
R-SEC-03	La solution supporte l'authentification selon le standard SAML2

<b>R- SEC -04</b>	<b>Les applications WEB respectent les règles de sécurité OWASP</b>
<b>R- SEC -05</b>	La solution doit mettre à disposition des points de monitoring (sous forme d'API REST ou autre à définir)

## 9 Informatique utilisateurs

Matériel et logiciel standard (non spécifique à un métier) mis à disposition des collaborateurs.

### 9.1 Matériel

Catégorie	Standards actuels	Commentaires
<b>Tablette</b>	Samsung Tab Active 2	Mis à jour le 01.01.2020

### 9.2 Systèmes d'exploitation

Catégorie	Standards actuels	Commentaires
<b>Poste de travail fixe</b>	Windows 10	Encore supporté : <ul style="list-style-type: none"> <li>Windows 7 SP1</li> </ul>
<b>Poste de travail portable</b>	Windows 10	Encore supporté : <ul style="list-style-type: none"> <li>Windows 7 SP1</li> </ul>
<b>Station de travail</b>	Windows 10	Encore supporté : <ul style="list-style-type: none"> <li>Windows 7 SP1</li> </ul>
<b>Tablette</b>	Android	

### 9.3 Logiciels

En plus des logiciels préinstallés, les logiciels ci-dessous sont supportés.

Catégorie	Standards actuels	Commentaires
<b>Antivirus</b>	Sentinel One	Mis à jour le 11.10.2018
<b>Suite bureautique</b>	Microsoft Office 365	Mis à jour le 11.10.2018
<b>SAP</b>	SAPGUI	Mis à jour le 11.10.2018
<b>Communications unifiée</b>	Avaya One-X Communicator Skype Entreprise	
<b>Connexion à distance</b>	Team Viewer	Encore supporté : <ul style="list-style-type: none"> <li>Ultra VNC Viewer</li> </ul>

<b>Navigateur</b>	Mozilla Firefox ESR	Dernière version ESR
<b>Imagerie</b>	Gimp	Mis à jour le 11.10.2018
<b>Lecteur Vidéo</b>	VLC	Mis à jour le 11.10.2018
<b>Gestion projets</b>	Microsoft Project GanttProject	Mis à jour le 11.10.2018
<b>Mindmapping</b>	FreeMind	Mis à jour le 11.10.2018
<b>Dessin technique</b>	Autocad +	
<b>Processus</b>	Process4BIZ	
<b>Dessin</b>	Microsoft Visio Photoshop Adobe Illustrator Draw.io	
<b>Texte</b>	NotePad++	Mis à jour le 11.10.2018
<b>Lecteur PDF</b>	PDF Xchange Editor	Mis à jour le 11.10.2018
<b>Visionneur AutoCad</b>	DWG TrueView	Mis à jour le 11.10.2018
<b>Visionneur HPGL</b>	HPGL Viewer	Ajouté le 01.07.2016

## 10 Informatique mutualisée

### 10.1 Matériel et infrastructure

Catégorie	Standard actuels	Commentaires
<b>Imprimante</b>	Dialogue : PostScript	
<b>Serveur</b>	HP Proliant DL360 Gen10 Intel	HP Proliant DL360 Gen8-9 Intel Mis à jour le 11.10.2018
<b>Téléphonie</b>	Avaya Aurora	
<b>Réseau wifi</b>	AP : HP 560 Wireless 802.11ac AP HP MSM466 Contrôleurs : HP 850 Unified Wired-WLAN Appliance	Mis à jour le 13.01.2017
<b>Réseau LAN</b>	Access : HP 5120 + HP 5130 Core : HP 5920	Mis à jour le 09.01.2017

	MikroTik PowerBox Moxa IKS-6728-8Poe A Moxa EDS-P510	
<b>Firewall</b>	Se référer au responsable de l'infrastructure SI	Mis à jour le 09.11.2018
<b>Load Balancer / Proxy</b>	F5 (Apache en décommissionnement)	Mise à jour juin 2019
<b>VPN</b>	Natif Windows 10 Plateforme SSL PulseSecure	Mis à jour le 09.01.2017
<b>Radio</b>	Switch Tetra : Etelm v2. Station de base : Etelm v2. Console Tetracs : Etelm v2. Terminaux : Motorola MTH800 Terminaux MDT 400 Teltronic FH Sratex.	Mis à jour le 16.01.2017
<b>Tablette</b>	Airwatch	Gestion du déploiement et gestion des app collaborateurs à distance et de manière centralisée.  Mis à jour le 01.01.2020

## 10.2 Systèmes d'exploitation, virtualisation et container

Catégorie	Standards actuels supportés	Commentaires
<b>Système d'exploitation</b>	OpenSuse Leap Windows Server 2016	SLES seulement pour environnement avec support fournisseur (SAP, Alfresco)  Encore supportés : <ul style="list-style-type: none"> <li>Windows Server 2012</li> <li>Windows Server 2012 R2</li> <li>Windows Server 2008 R2</li> <li>Windows Server 2008</li> </ul>
Virtualisation	VMWare 6.5	
Container	Docker	

Les règles suivantes sont appliquées afin de faciliter l'évolution des solutions.

Référence	Description
R-OS-01	La solution s'exécute sur système d'exploitation Linux et/ou Windows, tels que définis ci-dessus ; la préférence sera donnée à une solution qui ne dépend pas du système d'exploitation et qui peut être indifféremment déployée sur l'un ou l'autre.
R-OS-02	Lorsque la solution est hébergée par tl, elle s'exécute sur environnement virtualisé VMWare.
R-OS-03	Lorsque la solution est hébergée et exploitée par tl, elle est livrée sous forme de package Docker.

### 10.3 Logiciels

Catégorie	Standard actuels	Commentaires
Téléphonie	Avaya Aurora 6 One-X Agent 2.5 One-X communicator 6.2 One-X attendant 3.07	
Backup	NetBackup Snap Creator Framework	
Créa images	Paragon Software Free	
Base de données	MariaDB MS SQL Server MongoDB (embarqué)	DB Hana pour les développements dans SAP (MaxDB encore supporté). Redshift pour les développements Cloud. MySQL et Oracle encore supportés
SAP	NetWeaver 7 ERP 6.0 Ehp 8	
Déploiement postes de travail	SCCM WDS avec MDT	
Antivirus	Sentinel One	
Ticketing	GLPI	
Annuaire LDAP	AD (WS 2012 R2) Azure	
CMS	Joomla	
GED	Alfresco	

<b>SIG</b>	MapInfo	
<b>Monitoring</b>	Nagios	

Les règles suivantes sont appliquées afin de faciliter l'évolution et l'exploitation des solutions.

Référence	Description
<b>R-SOFT-01</b>	Lorsque la solution est exploitée par tl, elle s'appuie sur une base de données relationnelle supportée par tl. La préférence sera donnée à une solution qui ne dépend pas d'un moteur particulier de base de données.
<b>R-SOFT-02</b>	

## 10.4 Conventions de nommage

### 10.4.1 API

Les ressources REST sont exposées avec un URL déterminé comme suit :

- le préfixe est <https://api.t-l.ch>
- suit le nom de domaine métier (par exemple rh, finance, transport, iv, ...)
- suit un éventuel nom de sous-domaine
- se termine par le nom de la resource

Par exemple :

- <https://api.t-l.ch/rh/collaborateur> : resource pour les opérations sur un collaborateur
- <https://api.t-l.ch/transport/vehicule> : resource pour les opérations sur un véhicule

Le nom de domaine et de la resource sont obligatoires.

### 10.4.2 Queues et Topics

Le nom des queues et topics pour les échanges de messages asynchrones sont déterminés comme suit :

- Le préfixe est /TL
- Suit le nom du domaine métier (par exemple rh, finance, transport, iv, ...)
- suit un éventuel nom de sous-domaine
- suit une classification selon les besoins à adresser, propres à chaque cas d'utilisation

Par exemple :

- /TL/rh : événements propres au domaine des ressources humaines
- /TL/IV/noVhc/123/InfoVHC/Config : événements propres à la configuration d'un véhicule

Les deux premiers niveaux sont obligatoires, les autres en fonction du besoin.

### 10.4.3 Serveurs

La règle de nommage d'un serveur est comme suit :

Type	Environnement	Réseau	Nom Application	Numéro instance	Lieu
srv	p	e	XXXXX	01-99	bor
	t	i			per
	d				
	b				

Environnements :

- p : production
- t : test
- d : développement
- b : bac à sable

Réseau :

- e : réseau exposé à l'externe (DMZ)
- i : réseau interne

Nom Application

- abréviation sur au maximum 5 caractères

Numéro instance :

- Pour plusieurs de la même application, numérotation sur 2 chiffres de 01 à 99

Lieu (uniquement pour serveur physique) :

- bor : datacenter de La Borde
- per : datacenter de Perrelet

Exemple ;

srvpiscm01 : serveur (srv) de production (p) en réseau interne (i) pour application sscm - première instance (01)

## 11 Environnement de développement et d'intégration – Gestion de projet

La liste des logiciels ci-dessous fait référence pour les développements effectués par les équipes tl ou pour des développements spécifiques réalisés pour tl.

Catégorie	Standard actuels	Commentaires
<b>Développement UI</b>	TypeScript Angular et React	Fiori pour les développements dans SAP JavaScript + Flow encore supporté mais pas à développer.
<b>Mobile Devices</b>	Ionic	
<b>Développement Backend</b>	Java (OpenJDK) - Spring	



	NodeJS – Express	
Code Quality	Sonar	
Build et gestion des dépendances	Maven NPM	
Web services	REST + Swagger pour la documentation	Format par défaut JSON Format OData pour SAP
Gestion des sources	Git CodeCommit (AWS)	
ETL	Talend	
ESB	Talend	
Message Broker	ActiveMQ Mosquitto (MQTT - embarqué)	
BI Report	Tableau	Cognos en cours de décommissionnement
Intégration Continue	Jenkins / Nexus	
Serveur Applicatif	Tomcat	
Serveur web	Apache	
Statistiques	R	
Scénario de test	Jira	
Gestion de projet et gestion de tâches	Jira / Jira Agile	
Wiki projet	Confluence	

Les règles suivantes sont appliquées afin de faciliter l'évolution et l'exploitation des solutions.

Référence	Description
R-DEV-02	Lorsque la solution backend est développée ou reprise par les tl, la technologie utilisée est Java + Spring ou TypeScript + NodeJS.
R-DEV-03	Lorsque la solution frontend est développée ou reprise par les tl, la technologie utilisée est Angular ou React + Typescript.

