

Gestion des services dans une organisation en évolution

Novembre 2024

Présentation

- Qui est le présentateur?
 - Architecte de solutions chez Desjardins, Jean-Francois Pezet possède plus de 25 ans d'expérience en T.I. dans des entreprises de différentes tailles et dans différentes industries: des communications, des sciences de la vie, de la sécurité informatique et du secteur bancaire.
 - Desjardins : Plus de 52,000 employés dont plus de 6,000 en T.I.
- Qu'est-ce qu'un Architecte de solutions chez Desjardins?
 - Responsable de la solution
 - Qu'elle réponde aux besoins d'affaires
 - Qu'elle soit alignée sur les orientations de l'entreprise
 - Qu'elle respecte les règles de conformité et sécurité de l'entreprise

Une petite histoire ...

Petite histoire : évolution d'une entreprise

- Une PME
- Un propriétaire qui a des bonnes idées et des grandes ambitions!





Besoin : application de gestion des feuilles de temps

Contexte

- PME avec 5-10 employés

Besoins

- Application pour gérer les feuilles de temps pour le processus de paie et les réclamations pour recherche et développement

Utilisateurs internes

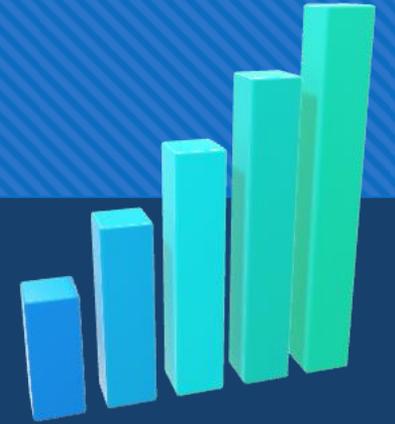


Application
Feuilles de
temps



BD
Feuilles de
temps

Besoin d'une autre application



Contexte

- L'entreprise grossit !
- Nouveaux employés à distance (vendeurs, installateurs)

Besoins

- Nouvelle application légère pour remplir les feuilles de temps

Utilisateurs internes



Application
Feuilles de
temps

Utilisateurs externes



Application
légère
Feuilles de temps



Nouvelle application de facturation

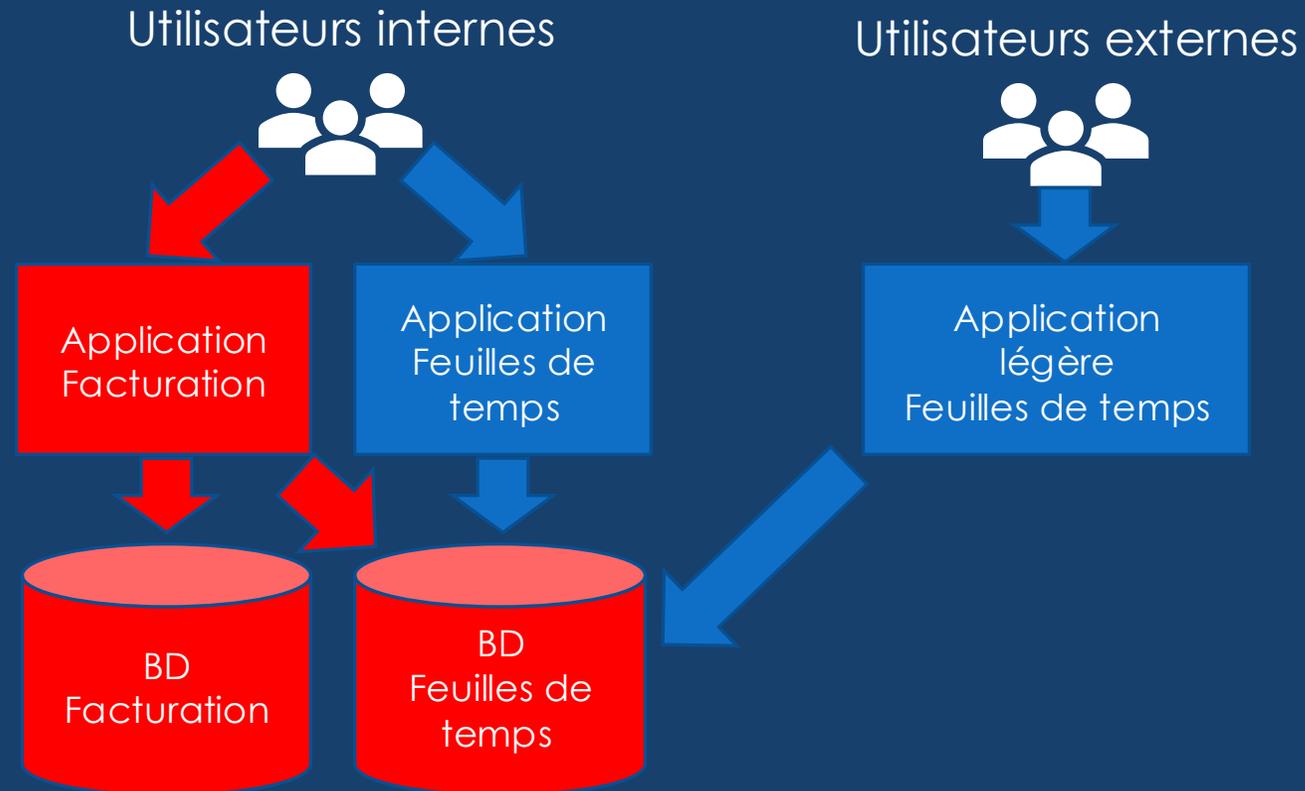


Contexte

- L'entreprise grossit !
- Dans 3-6 mois, arrivée d'une nouvelle application de Facturation qui aura besoin de l'information des feuilles de temps

Besoins

- Besoin de changer la BD donc, requiert changements dans toutes les applications



Croissance de l'entreprise



Contexte

- L'entreprise bonifie son offre de services
- L'entreprise offre ses services à des heures étendues alors la plage de maintenance diminue

Besoins

- Nouvelles applications à intégrer
- Nouveaux systèmes de différentes technologies à intégrer
- Évolution à prévoir aux BD
- Éviter d'affecter la continuité des affaires

Quoi faire?

- Ajouter un niveau d'abstraction : créer des services !

Les services

Terminologie

Service : regroupement de capacités fonctionnelles et données. Les traitements de ces capacités sont exécutés et encapsulés dans le service.

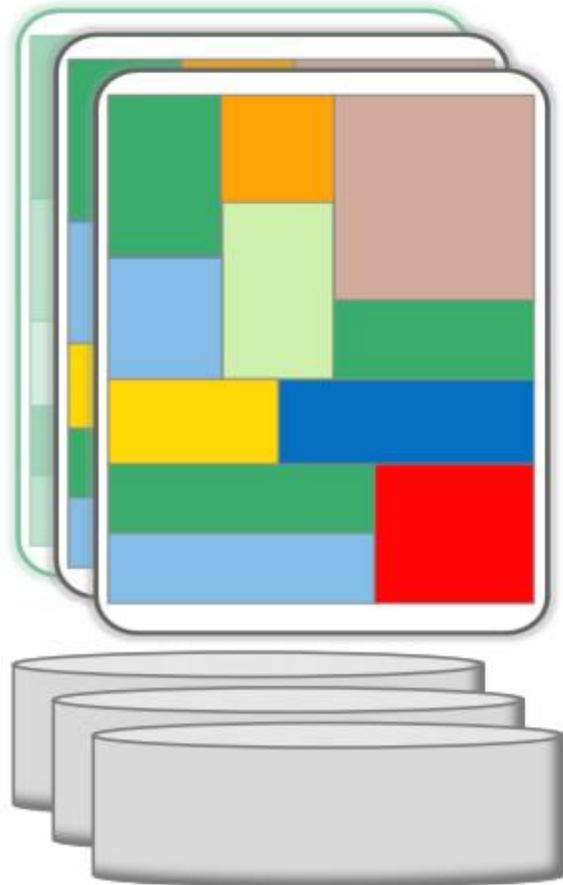
API : interface pour connecter des composants logiciels. Une API permet à une organisation d'exposer les données et fonctionnalités de leurs systèmes.

API Web : interface de logiciel exposée via le protocole HTTP/REST.

Microservice : composant indépendant qui offre une capacité ciblée et qui supporte une interconnexion par message. Un microservice est une spécialisation d'un service.

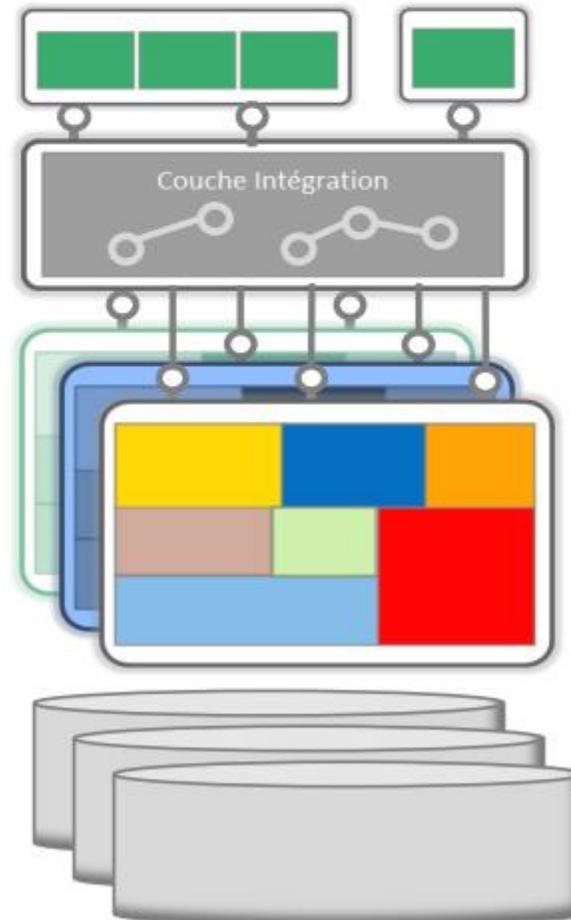
Architecture Monolithe

Les différents fonctionnalités d'affaires et les logiques de présentation sont regroupés dans un seul bloc d'application.
Focus: Contraintes de Projet



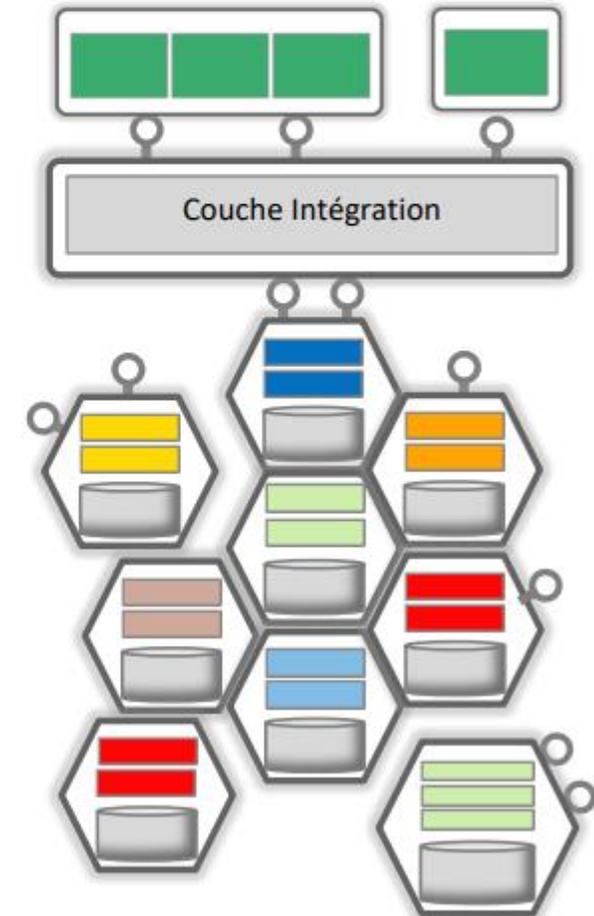
Architecture SOA 'Traditionnelle'

Un découpage par couche pour séparer les consommateurs et les fournisseurs de services responsables des différents fonctionnalités d'affaires.
Focus: Réutilisation des Services



Architecture SOA 'Microservices'

Les fonctionnalités d'affaires conçues dans des services autonomes (exécution et données) sont composés pour bâtir une application.
Focus: Agilité de livraison



Principes SOA

SOA = Architecture Orientée Service

-Abstraction

-Sans couplage

-Réutilisation

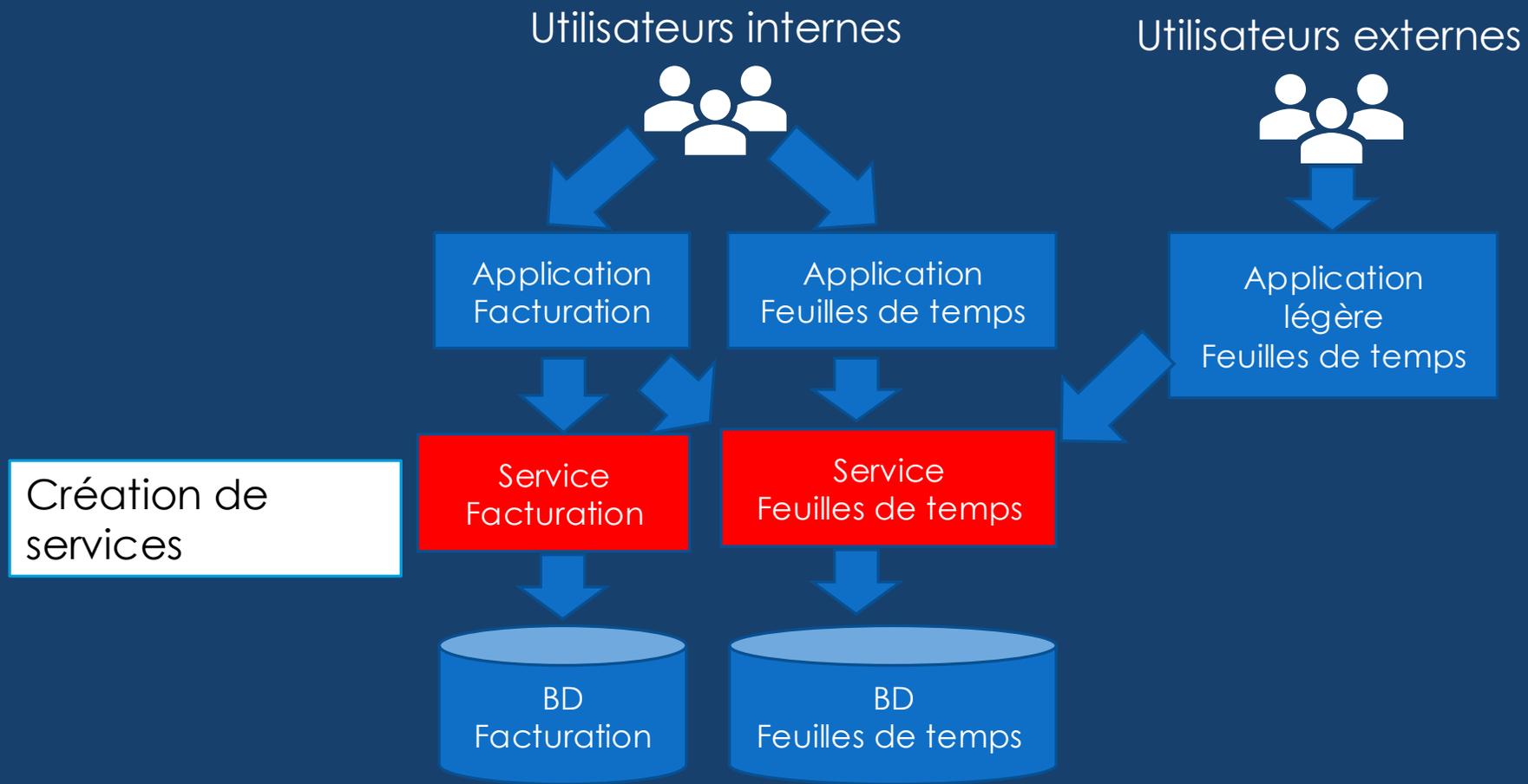
-Autonome

-Sans-état

-Contrat standardisé

-Peut être découvert

-Composition



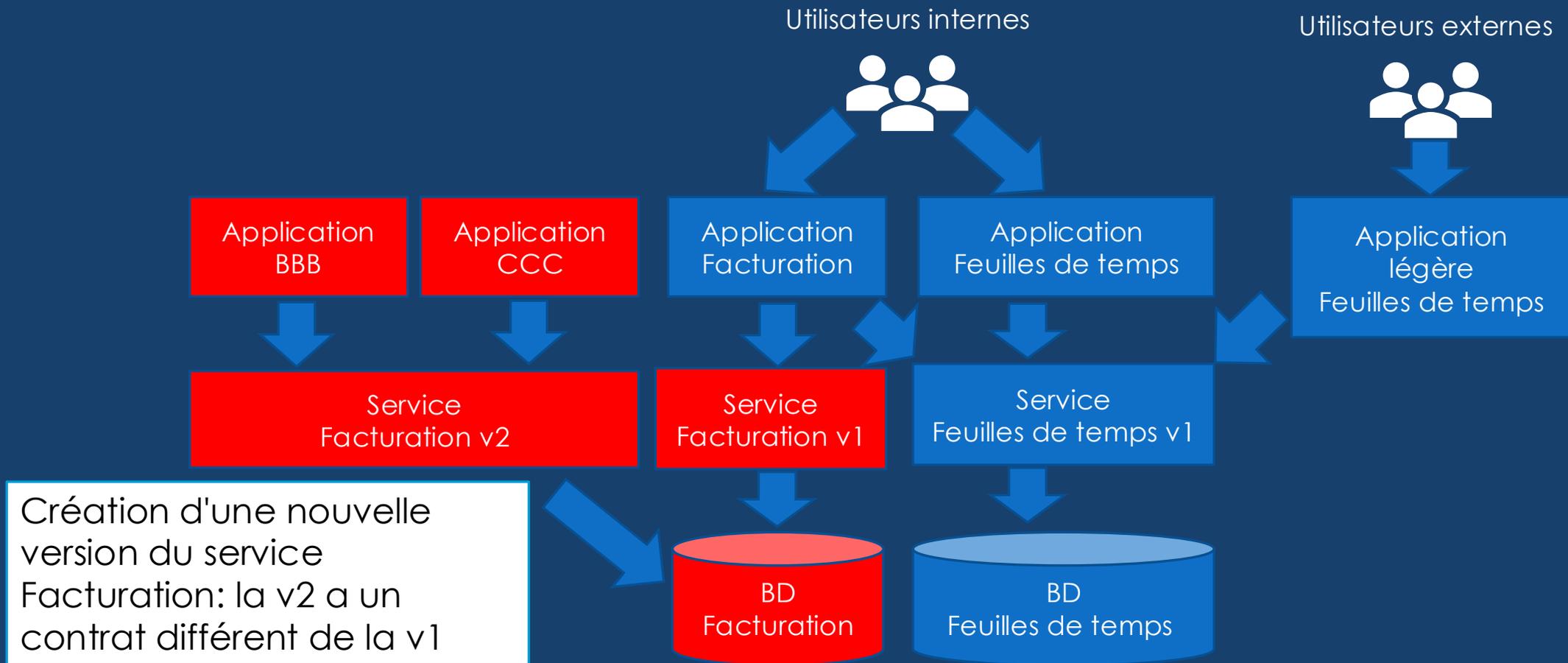
Nouveaux systèmes à intégrer

Contexte

- L'entreprise bonifie son offre de services

Besoins

- Nouveaux systèmes qui ont besoin de s'intégrer à la BD Facturation et qui ont besoin de faire modifier la BD Facturation
- Continuité des affaires : on ne veut pas changer l'application de Facturation car elle est trop critique aux affaires



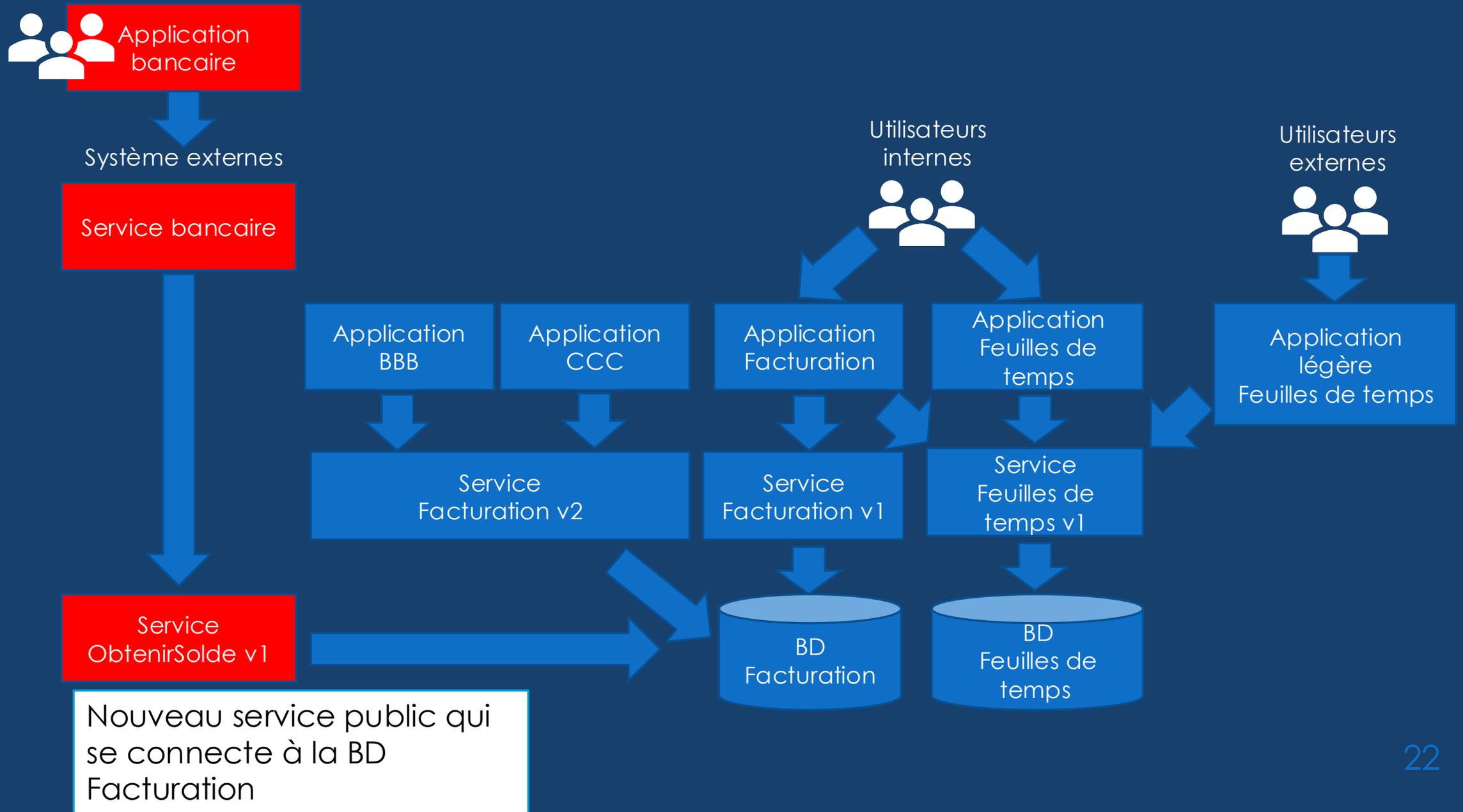
Service externe répondant à un standard

Contexte

- Un nouveau service bancaire est offert pour permettre aux membres/clients de considérer leur solde en souffrance chez leurs fournisseurs (ex : mes dettes envers Hydro-Québec, Vidéotron, fournisseur ABC, etc.)
- L'utilisation de ce service est importante pour donner un avantage concurrentiel à l'entreprise

Besoins

- Permettre aux entreprises de faire partie de ce service dans les applications libre-service bancaires (ex: AccèsD) – il faut exposer un service externe répondant à un standard (ObtenirSolde(id client, date, ...))
- Le solde des comptes clients se trouve dans la BD facturation
- Il faut considérer la continuité des affaires



Architecture *Microservices*

Architecture microservice

- Un **microservice** est un composant indépendant qui offre une capacité ciblée et qui supporte une interconnexion par message
- Une **architecture microservice (MSA)** est un style d'architecture hautement automatisé, évolutif et composé de microservices

Architecture microservice

- Une spécialisation du SOA
- Un microservice doit encapsuler sa logique d'affaires et ses données
- Décentralisée
- Avantages ?
 - Meilleur découpage conceptuel
 - Conception, développement, test et déploiement en isolation
 - Plus rapide et facile à développer, tester et déployer
 - Plus facile à opérationnaliser (stabilité, élasticité)

Architecture microservice

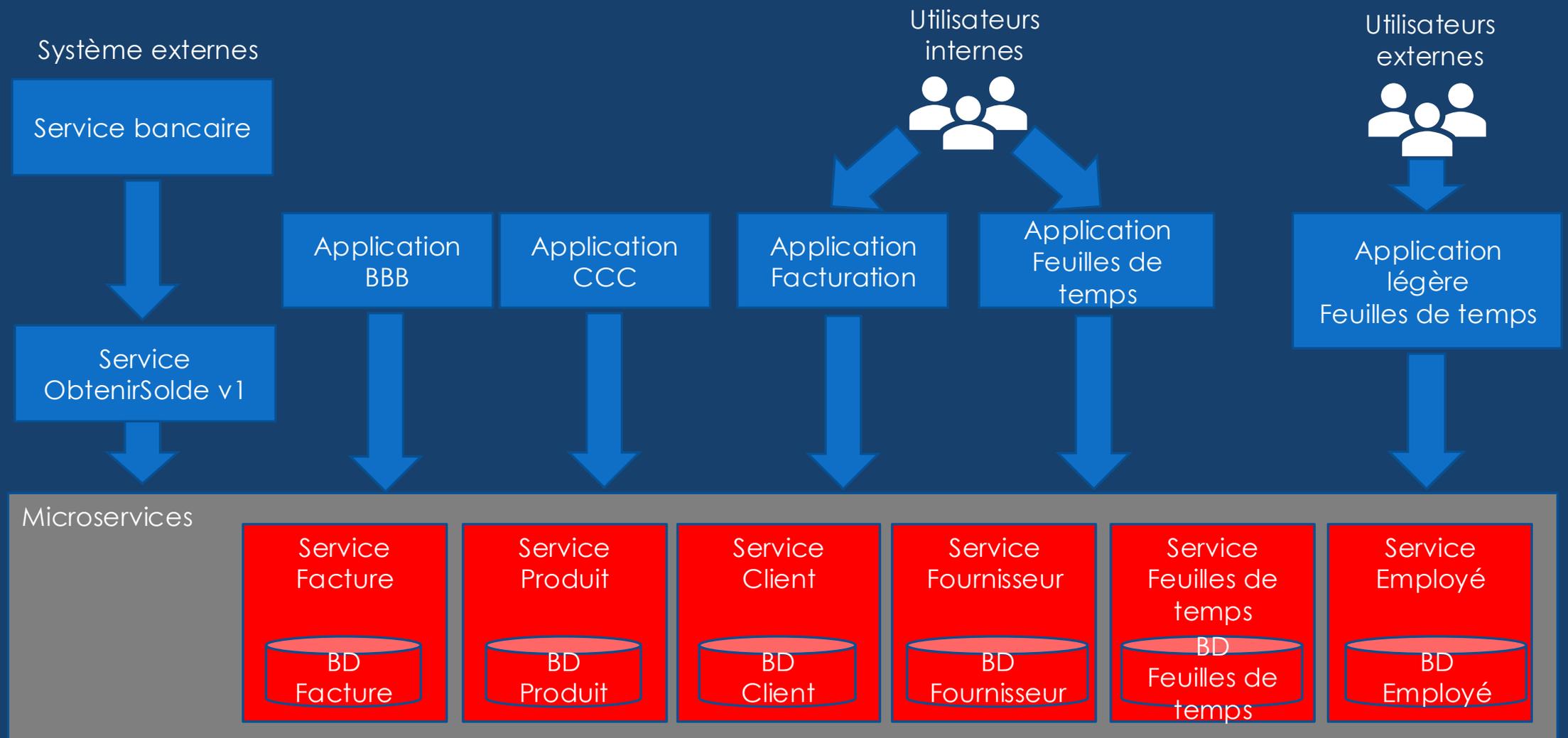
Caractéristiques d'un microservice (Martin Fowler)

- Un composant est un service
 - Composant implémenté en service (et non en librairie) qui peut être remplacé par un autre
- Organisé autour des capacités d'affaires
 - Organisation d'équipe multidisciplinaire par capacité d'affaire (et non par spécialité technique)
- Approche produit et non projet
 - Gestion du service par produit (longue durée) et non par projet (durée limitée)
- Interfaces intelligentes et plomberies sans intelligence
 - Réseau simpliste et service intelligent (éviter les réseaux intelligents (ESB))

Architecture microservice

Caractéristiques d'un microservice (Martin Fowler)

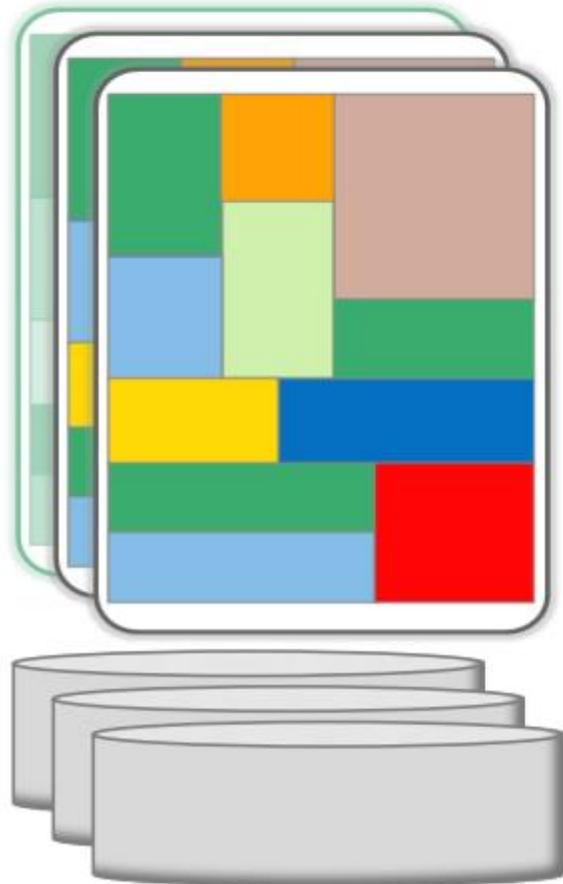
- Gouvernance décentralisée
- Gestion de données décentralisée
 - Chaque service a ses propres données (et sa propre technologie de gestion de données) et ne les partage pas
- Automatisation (test, déploiement, infrastructure)
 - Déploiement continue, rapide, fréquent, sans perte de service et avec retour arrière facile
- Conception pour supporter défaillances et interruptions
 - Planifier en conséquence car le service va échouer, les erreurs vont survenir!
- Conception évolutive



Caractéristiques des architectures

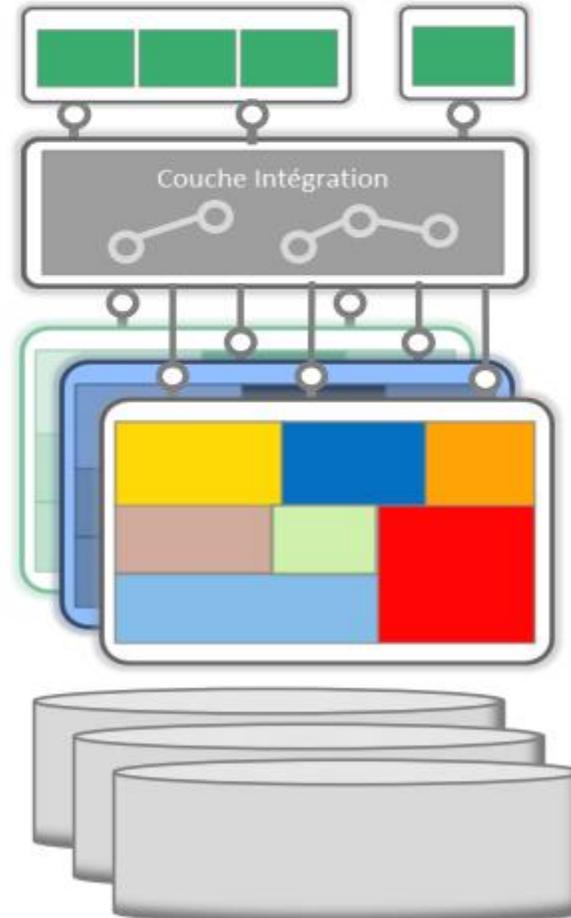
Architecture Monolithe

Les différents fonctionnalités d'affaires et les logiques de présentation sont regroupés dans un seul bloc d'application.
Focus: Contraintes de Projet



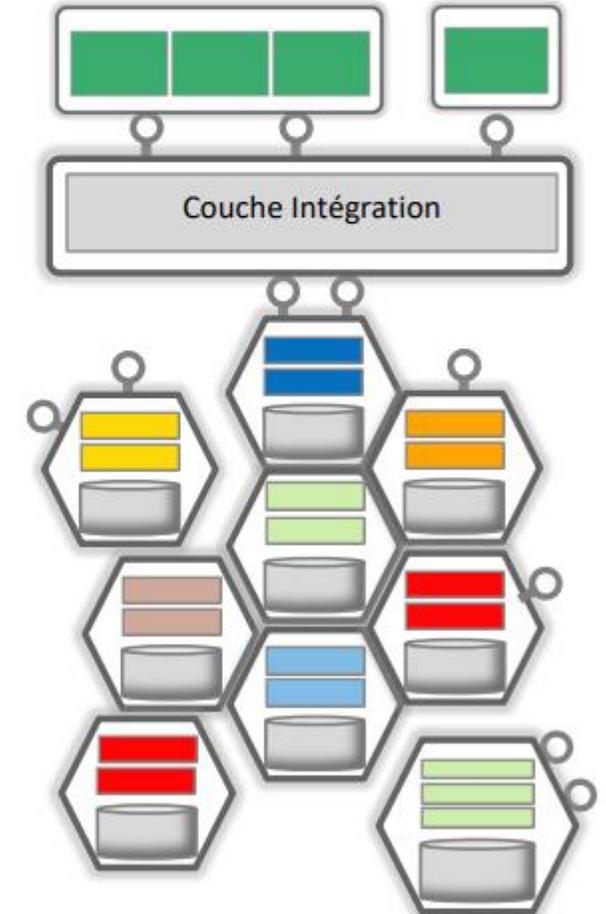
Architecture SOA 'Traditionnelle'

Un découpage par couche pour séparer les consommateurs et les fournisseurs de services responsables des différents fonctionnalités d'affaires.
Focus: Réutilisation des Services



Architecture SOA 'Microservices'

Les fonctionnalités d'affaires conçues dans des services autonomes (exécution et données) sont composés pour bâtir une application.
Focus: Agilité de livraison



Un autre niveau de croissance de l'entreprise

Contexte

- Augmentation du nombre de systèmes à intégrer : de 5 à 50 !!!
- Augmentation du nombre d'utilisateurs de 10 à 500 !!!
- Augmentation de la charge sur les serveurs : de 10 transactions/minute à 1,000 transactions/minute !!!

Besoins

- Comment supporter l'augmentation de la charge ?
- Comment contrôler les accès ?
- Comment assurer le niveau de sécurité requis par nos partenaires externes ?
- Comment remplir les besoins de conformité : qui a fait quoi quand ?
- Comment faire tout cela en tenant compte de la continuité des affaires ?

Critères à considérer lors de la conception d'un service

- Contexte d'utilisation
 - Quels sont les cas d'utilisation du service ?
 - Interne, ouvert au public, ouvert à des partenaires
- Besoins non-fonctionnels
 - Performance, temps réponse
 - Sécurité
 - Traçabilité, journalisation
 - Volumétrie
 - Synchrone, asynchrone
- Besoins opérationnels
 - Plage de maintenance
 - Niveau de service requis
- Évolution des capacités d'affaires
 - Facilité d'évolution

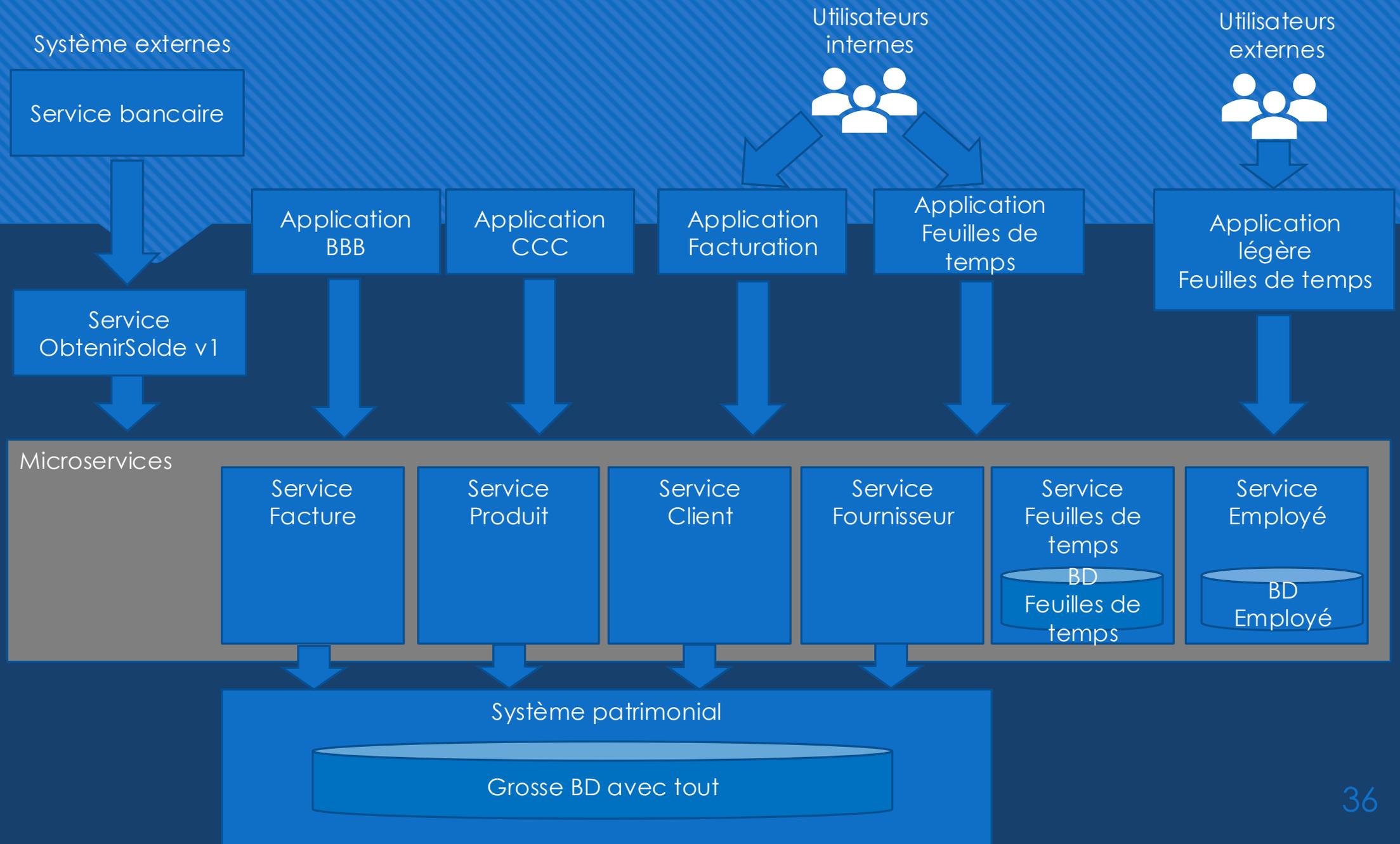
Défis de l'architecture microservice

Défis de l'architecture microservice

- Équipes décentralisées
- Perte de vision sur la solution entière car le focus est sur une petite capacité très ciblée
- Plus difficile de trouver une anomalie dans une solution car elle est très décentralisée
- Le coût de l'enrobage est plus grand que le coût de développement de la capacité fonctionnelle
 - Enrobage : mise en place, gouvernance, sécurité, opérationnalisation, journalisation, etc.
- Impact sur le réseau : plusieurs petits appels de service

Défis de l'architecture microservice

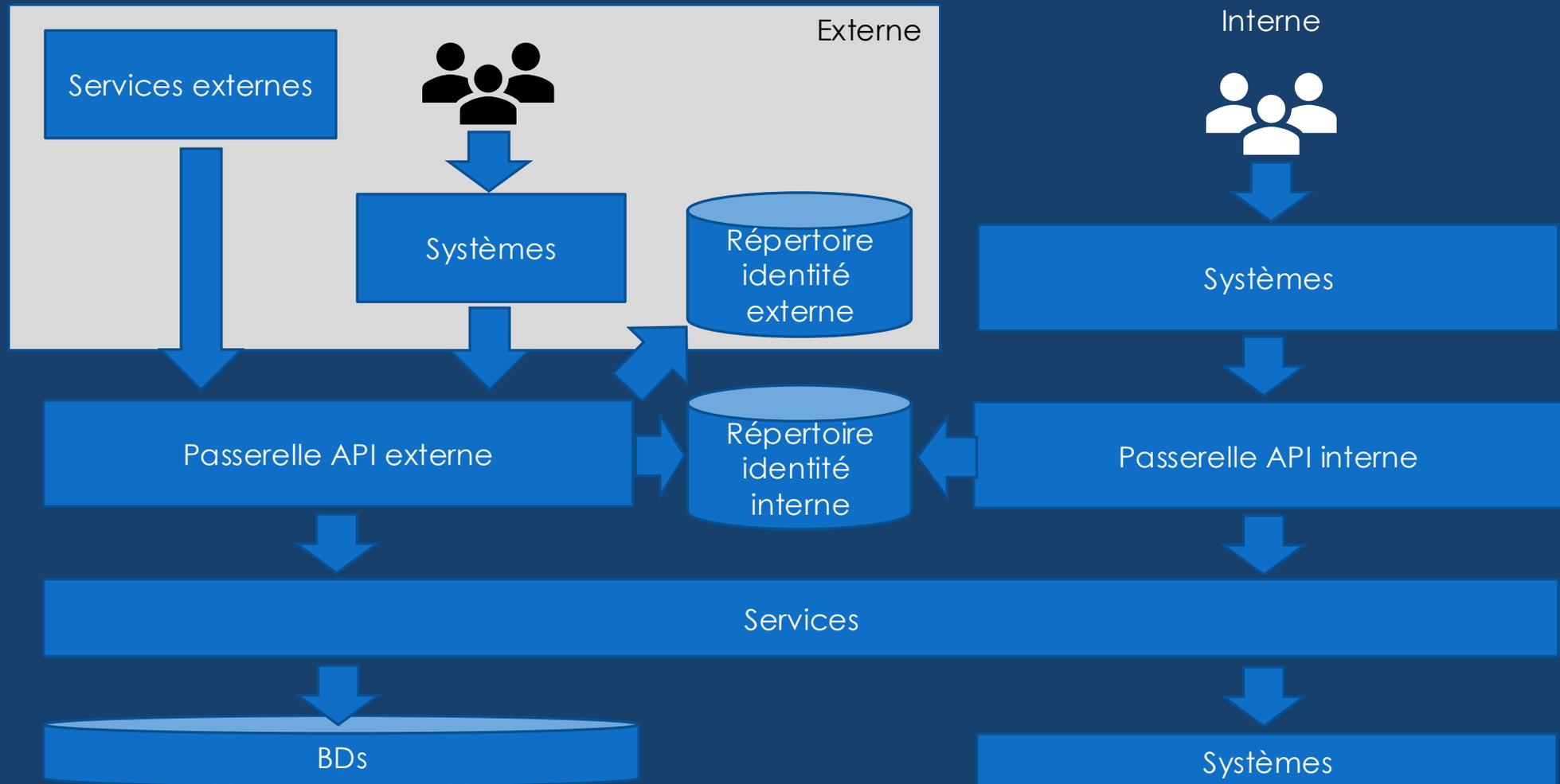
- Qu'est-ce qui arrive lors de l'intégration à des systèmes patrimoniaux?
 - Est-ce qu'on conserve les avantages de l'architecture?
 - Est-ce efficace?



Mise en place des services

Un modèle (parmi tant d'autres)

- Utilisation d'un niveau d'abstraction réseautique (ex: passerelle API)
 - Rediriger les requêtes vers le bon fournisseur de service
 - Appliquer la sécurité du service (authentification et autorisation)
 - Effectuer la journalisation et la surveillance
 - Apporter une abstraction du fournisseur de service (Java, .NET, cloud, serveurs internes, etc.)
- Le service (fournisseur de service)
 - Respecter le contrat de service
 - Appliquer les règles d'affaires
 - Appliquer la sécurité métier (ex : accès à des données selon un rôle)
- Balance de charge
 - Chaque composant a son balancer de charge : réseau, fournisseur de service, BD, etc.



Impacts organisationnels

Complexité des systèmes : plus de services!

- De 2 services à 200 services
- Qui doit créer les services?
- Comment retrouver l'information sur les services?
- Comment éviter la duplication?

Création des services

- Dans la majorité des cas, on crée des services pour répondre à des besoins d'affaires, donc les services doivent cadrer dans une architecture d'affaires : quels sont les domaines d'affaires de l'entreprise?
- La conception des services doit se faire avec les gens d'affaires

Création des services – qui ?

- Centralisation - Équipe transversale : Une seule équipe crée les services pour toute l'entreprise
 - Avantage : plus facile d'assurer une constance dans les services et éviter la duplication
 - Désavantages : demande aux gens d'informatique de connaître plusieurs secteurs d'affaires
 - En général, convient mieux à une petite/moyenne entreprise
- Décentralisation - Équipe par secteur d'affaires (ou métier)
 - Avantage : l'équipe est plus près des gens d'affaires et devient des spécialistes métier
 - Désavantages : plus difficile d'assurer la constance et d'éviter la duplication
 - En général, convient mieux à une grosse entreprise

Les services en fonction de deux perspectives

- Perspective Affaires
 - Un service offre une nouvelle façon de s'ouvrir à l'externe et ainsi augmenter ses revenus. Les services peuvent être des nouveaux canaux de distribution permettant des nouveaux modèles d'affaires. Les services sont essentiels au virage numérique de l'organisation.
- Perspective Technologique
 - Un service permet d'accéder aux fonctionnalités et aux données de façon simple et flexible.

Situation

Situation

- L'équipe Facturation crée des services de facturation en incluant l'information sur l'adresse des clients
- L'équipe Compte Client crée des services de gestion de compte en incluant l'information sur l'adresse des clients

Questions

- Est-ce que l'information sur l'adresse des clients est définie de manière identique dans les deux familles de services?
- Quelle est la source autoritaire de cette information?
- Est-ce que la gestion des informations des clients devrait avoir sa propre famille de services?

Gouvernance

- Répertoire corporatif des services
 - Site Web, UDDI, outil spécialisé ou autre
 - Sert à répertorier les services avec toutes informations pertinentes (contrat, disponibilité, historique des versions, propriétaire, niveau de service, etc.)
- Taxonomie des services
 - Devrait refléter l'architecture d'affaires (le classement doit avoir du sens pour toute l'organisation)
 - Il existe des standards par industrie (ex : BIAN - Banking Industry Architecture Network)
- Nomenclature des services
 - Utiliser un langage commun (selon un modèle conceptuel d'information de l'entreprise)
 - Règles pour la nomenclature
 - des services
 - des opérations
 - des paramètres

Comité de gouvernance

- Dans les grandes organisations, un comité peut être mis en place pour assurer la gouvernance des services.
- Exemple: « API Design Authority » qui s'assure ...
 - Que les gens d'affaires sont impliqués dans la conception des services et que le service a une grande valeur d'affaires pour l'organisation
 - Que la taxonomie du service est bien respectée
 - Que l'architecture du service permet de rencontrer les besoins non-fonctionnels :
 - Sécurité, performance, etc.
 - Que les lignes directrices et les standards sont respectés

Taxonomie des services

Taxonomie des services

- Il existe plusieurs façons de classer les services ...

Exemple : le BIAN

BIAN Service Landscape - V6.0

Reference Data	Sales & Service		Operations & Execution			Risk & Compliance	Business Support	
Party	Channel Specific	Marketing	Product Specific Fulfillment			Bank Portfolio & Treasury	IT Mngmt.	Finance
Party Data Mngmt. Customer Profile	Branch Location Mngmt. Contact Center Mngmt. Branch Network Mngmt. E-Branch Mngmt. Adv. Voice Services Mngmt. ATM Network Mngmt. Contact Center Operations Branch Location Operations E-Branch Operations Adv. Voice Services Operat. ATM Network Operations Branch Currency Mngmt. Branch Currency Distribution Prod. Inventory Item Mngmt. Prod. Inventory Distribution Card Terminal Card Terminal Operation	Business Development Brand Mngmt. Advertising Promotional Events Prospect Campaign Mngmt. Prospect Campaign Design Customer Campaign Mngmt. Customer Campaign Design Customer Surveys	Loans & Deposits	Investment Mngmt.	Trade Banking	Corporate Treasury Analysis Corporate Treasury Asset Securitization Asset & Liability Mngmt. Bank Portfolio Analysis Bank Portfolio Administration Stock Lending/Repos	IT Systems Direction IT Stds & Guidelines Systems Administration Development Environment System Development Production Release System Deployment Systems Operations Platform Operations Systems Help Desk Systems Assurance Internal Network Operation	Financial Statements Financial Control Financial Compliance Enterprise Tax Administration
External Agency			Loan Leasing Current Account Deposit Account Corporate Current Account Consumer Loan Corporate Loan Corporate Deposits Corporate Lease Merchandising Loan Mortgage Fiduciary Agreement Savings Account	Investm. Portfolio Planning Investm. Portfolio Analysis Investm. Portfolio Mngmt. eTrading Workbench	Letter of Credit Bank Guarantee Trade Finance Credit Mngmt. Credit Facility Project Finance Limit & Exposure Mngmt. Syndicated Loan Cash Mngmt. & Account Services Direct Debit Mandate Direct Debit Cheque Lock Box Factoring	Models	Human Resource Mngmt.	
Market Data	Cross Channel	Sales	Cards	Wholesale Trading	Corp. Financing & Advisory Services	Market Risk Models Financial Inst. Valuation Models Gap Analysis Credit Risk Models Liquidity Risk Models Economic Capital Business Risk Models Customer Behavior Models Fraud Models Credit/Margin Management Production Risk Models Operational Risk Models Contribution Models	Non IT & HR Enterprise Services	Knowledge & IP Mngmt.
Information Provider Operation Market Information Mngmt. Financial Market Analysis Financial Market Research Quant Model Market Data Switch Admin Market Data Switch Ops Financial Instr. Ref Data Mngmt. Counterparty Administration Public Reference Data Mngmt. Location Data Mngmt.	Party Authentication Transaction Authorization Point of Service Servicing Event History Servicing Activity Analysis Contact Routing Contact Dialogue Interactive Help Contact Handler Customer Workbench	Prospect Campaign Execution Prospect Mngmt. Lead/Opportunity Mngmt. Customer Campaign Execution Customer Offer Sales Planning Underwriting Commission Agreement Commissions Product Matching Product Expert Sales Support Product Sales Support Sales Product	Credit/Charge Card Card Authorization Card Capture Card Billing & Payments Merchant Relations Merchant Acquiring Card Network Participant	Trading Book Oversight Trading Models Dealer Workbench Quote Mngmt. Suitability Checking Credit Risk Operations Market Making ECM / DCM Program Trading Traded Position Mngmt. Market Order Market Order Execution	Corporate Finance M&A Advisory Corp. Tax Advisory Public Offering Private Placement	Market Risk Models Financial Inst. Valuation Models Gap Analysis Credit Risk Models Liquidity Risk Models Economic Capital Business Risk Models Customer Behavior Models Fraud Models Credit/Margin Management Production Risk Models Operational Risk Models Contribution Models	Legal Compliance Internal Audit Security Advisory Security Assurance Approved Supplier Directory Procurement Company Billing & Payments Fixed Asset Register	Mngmt. Manual Intellectual Property Portfolio Knowledge Exchange
Product Management		Customer Mngmt.	Consumer Services	Market Operations		Business Analysis	Buildings, Equipment and Facilities	Corporate Relations
Product Design Product Deployment Product Training Product Quality Assurance Discount Pricing Product Directory Special Pricing Conditions		Customer Relationship Mngmt. Customer Prod./Service Eligibility Customer Agreement Sales Product Agreement Customer Access Entitlement Customer Behavioral Insights Customer Credit Rating Account Recovery Customer Event History Customer Reference Data Mgmt Customer Precedents Customer Proposition	Corporate Trust Services Remittance Currency Exchange Bank Drafts & Trvl. Checks Brokered Product Consumer Investments Customer Tax Handling Consumer Advisory Services Trust Services Service Product	Mutual Fund Admin. Hedge Fund Admin. Unit Trust Admin. Trade Confirmation Matching Order Allocation Settlement Obligation Mngmt. Securities Dlvry & Receipt Mngmt. Securities Fails Processing Trade/Price Reportg. Custody Administration Corporate Events Financial Instrument Valuation		Segment Direction Product Portfolio Customer Portfolio Branch Portfolio Channel Portfolio Competitor Analysis Market Research Market Analysis Contribution Analysis	Property Portfolio Site Operations Site Administration Equipment Administration Equipment Maintenance Utilities Administration Building Maintenance	Corporate Communications Corporate Alliance/Stakeholder Corporate Relationship Regulatory and Legal Authority Investor Relations
		Servicing	Cross Product Operations		Regulations & Compliance	Business Command & Control	Business Direction	
		Servicing Issue Customer Case Mngmt.			Guideline Compliance Regulatory Compliance Compliance Reporting Regulatory Reporting Fraud/AML Resolution Financial Accounting	Organization Direction Business Unit Financial Analysis Business Unit Financial Operations	Corporate Strategy Corporate Policies Product & Services Direction Business Architecture Continuity Planning	

Taxonomie technologique

- Données : gestion d'information (ex: client, facture)
- Transactionnel : exécuter une fonctionnalité transactionnelle (ex: ouvrir un compte, effectuer un transfert d'argent)
- Intégration : intégration entre des applications
- Interaction : contexte d'utilisation spécifique (ex: service d'interaction d'une page Web)
- Composite : composer plusieurs services pour produire un service
- Utilitaire : support technique (ex: journalisation, sécurité, conversion de format)

Taxonomie selon valeur d'affaires

- Produit / B2B : génère des revenus ou permet des échanges de données avec partenaire externe
- Solution / processus : spécifique à un processus, à une solution (ex: processus d'achat d'une voiture)
- Métier : spécifique à un domaine, maximum de réutilisation (ex: gestion des données du client)
- Intégration : intégration à un système spécifique (ex: connexion à un système patrimonial ou spécialisé)

Taxonomie

Utilisateurs internes



Application interne

Service Solution

Utilisateurs externes



Application externe

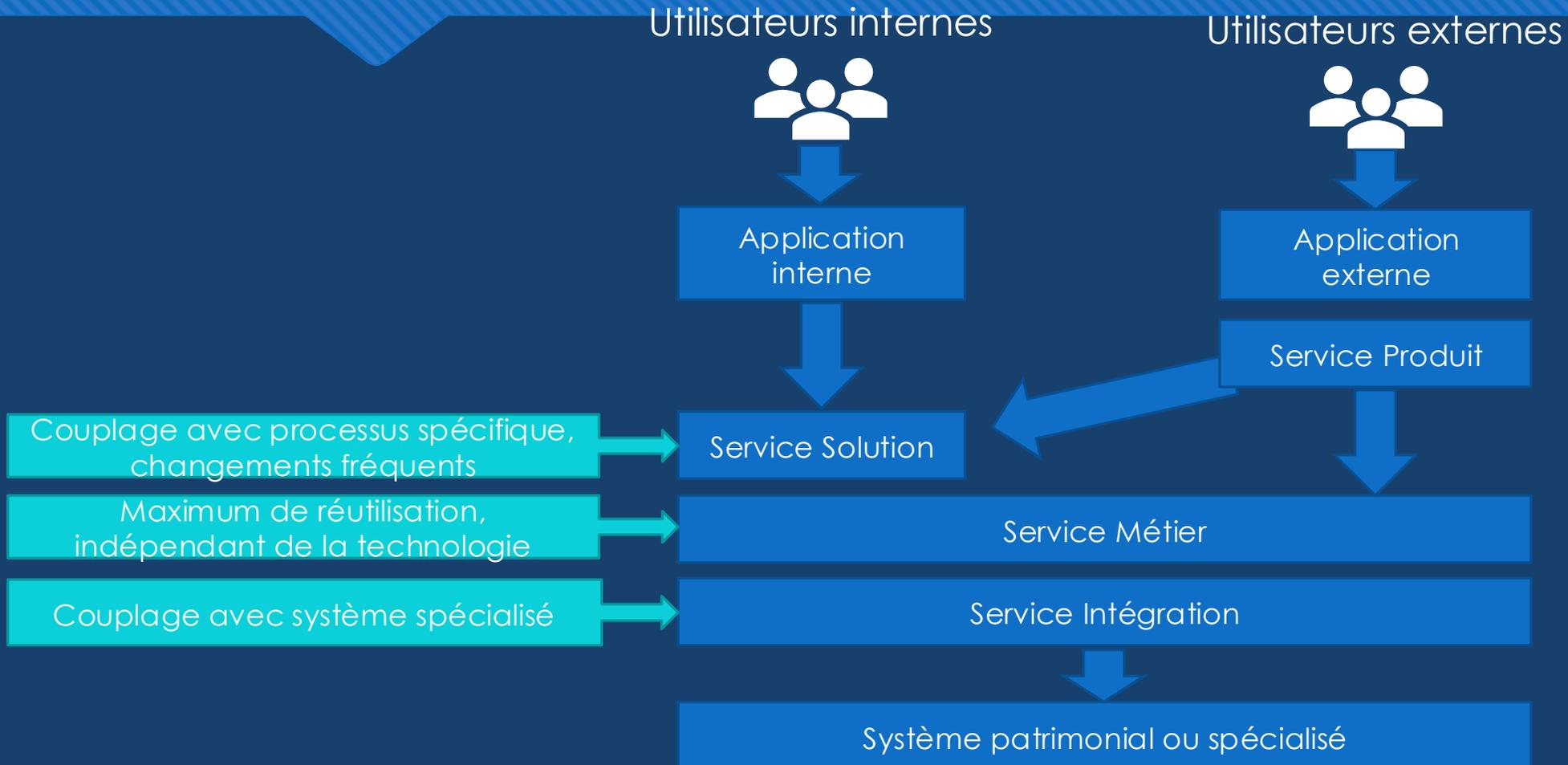
Service Produit

Service Métier

Service Intégration

Système patrimonial ou spécialisé

Taxonomie



Évolution des services

Évolution des services

- Propriétaire des services
 - Responsable des évolutions, des mises à jour et de la rétrocompatibilité des versions
 - Responsable de la gestion des versions des services
 - Responsable de s'assurer que l'infrastructure va supporter les requis non-fonctionnels (la charge, la disponibilité, le temps de réponse, etc.)

Utilisation des services

- Le projet consommateur annonce l'utilisation du service au propriétaire du service pour :
 - S'assurer d'avoir la plus récente information à propos du service (est-ce qu'une nouvelle version est en développement? Est-ce qu'on utilise le service de façon optimale pour répondre à notre besoin?)
 - S'assurer que les critères non-fonctionnels (incluant la nouvelle charge) seront supportés

Mesurer la performance des services

- Mesurer le temps réponse des appels de service
 - Utilisation de scénario d'appel de service prédéterminé (tests automatisés et répétitifs)
 - Gestion des balanceurs de charge (requêtes sur une seule passerelle API, un seul fournisseur de service, une seule BD, etc.)
 - Utilisation de simulateurs pour isoler la performance de certains composants (ex : simuler une BD)
 - Utilisation des « logs » lors des appels de service, idéalement avec outil d'agrégateur de « logs » (ex: Splunk)
 - Outil d'instrumentation des serveurs (ex : Dynatrace) sur tous les serveurs impliqués pour mesurer le temps réponse, l'utilisation des ressources (CPU, mémoire), taille des requêtes, etc.

Sécurité

- RBAC - Authentification / autorisation (lien avec répertoire d'identité)
- ABAC - Accès aux données
- Exposition des services à l'interne et à l'externe
- Cryptage des données
- Gestion des certificats

Surveiller les services

- Calculer l'utilisation des services : qui utilise les services et à quelle fréquence ...
 - Pour la facturation ?
 - Pour détecter des intrusions potentielles ?
 - Pour identifier les services et versions utilisées et inutilisées ?
 - Justifier le coût des services

Nouveaux besoins?

- Être plus agile
 - S'ajuster plus rapidement au marché
 - Ajouter des nouveaux canaux de distribution
 - Être apte à déployer plus fréquemment et plus rapidement
- Être plus élastique
 - Supporter plus facilement la croissance

Autres éléments à considérer

- Évolution des infrastructures (matériel, réseau, stockage)
- Évolution des systèmes (OS, serveur applicatif, RDBMS, Cloud)
- Changement des systèmes qui détiennent les sources autoritaires des données (ex : passer d'un système maison de facturation vers un ERP)
- La technologie change plus rapidement que les capacités d'affaires des entreprises : une entreprise pourrait toujours offrir un certain produit/service mais les systèmes informatiques requis pour le faire pourrait changer plusieurs fois dans la vie de l'entreprise

Gestion des versions d'un service

Gestion des versions d'un service

- Question : est-ce qu'il y a des impacts négatifs à créer des nouvelles versions d'un service ?

Gestion des versions d'un service

- Impacts ?
 - A partir du moment où elle est utilisée, une version d'un service peut vivre très longtemps
 - Chaque version doit être supportée
 - Gestion de code
 - Infrastructure
 - Correctifs de sécurité
 - Correctifs fonctionnels
 - Gestion des anomalies
 - Donc plusieurs versions à mettre à jour et à surveiller, etc.
 - La gestion des versions est un « anti-pattern » du Web : les consommateurs ne veulent pas et ne devraient pas savoir qu'il y a une nouvelle version (ex: lors d'un changement de version du navigateur)

Gestion des versions d'un service

- Une version majeure (ex: 2.0) devrait être utilisée pour refléter un changement qui ne peut pas garantir la rétrocompatibilité au contrat initial
- Une version mineure (ex: 1.1) devrait refléter un changement qui n'impacte pas les consommateurs courants

Gestion des versions d'un service

- Changez aussi souvent que vous voulez (version mineure) mais il faut **éviter les nouvelles versions majeures!**
- Comment éviter une version majeure ?
 - Toujours ajouter des capacités optionnelles
 - Ne jamais changer une capacité existante
 - Ne jamais enlever une capacité

Annexe

Terminologie

Service : regroupement de capacités fonctionnelles et données. Les traitements de ces capacités sont exécutés et encapsulés dans le service et sont rendus disponible par des opérations via une interface.

API : interface pour connecter des composants logiciels. Une API permet à une organisation d'exposer les données et fonctionnalités de leurs systèmes à travers des interfaces.

API Web : interface de logiciel exposée via le protocole HTTP/REST.

Fournisseur de service : instance qui contient et exécute le code du service

Continuité des affaires : la capacité d'une organisation à maintenir ou à reprendre rapidement les opérations de l'entreprise pour s'assurer d'offrir ses services et/ou ses produits avec un minimum d'interruption

RBAC (Role Based Access Control) : Contrôle des accès basés sur les rôles

ABAC (Attribute Based Access Control) : Contrôle des accès base sur des attributs

UDDI (Universal Description, Discovery and Integration) : standard XML pour décrire les services Web et permettre la découverte de ceux-ci

Principes agiles

Agile Manifesto Principles

- “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.”
- “Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.”

Les
microservices
apportent une
pression sur le
réseau ?

Michael T. Nygard

- *“Bugs will happen. They cannot be eliminated, so they must be survived instead.”*

Les microservices apportent une pression sur le réseau ?

Les patrons de stabilité de Nygard

- « Timeout » : arrêter d'attendre pour une requête qui n'arrivera jamais
- « Circuit Breaker » : coupure automatique du système lorsque celui-ci est sous pression
- « Bulkhead » : utilisation de balanceur de charge pour éviter qu'un problème affecte tous les consommateurs
- « Steady State » : le système devrait toujours fonctionner sans intervention humaine (nettoyage de ressources non-utilisées, stratégie de « caching »)
- « Fail Fast » : arrêter une opération si on sait d'avance que celle-ci va échouer
- « Handshaking » : permettre au composant appelé de limiter ses ressources (ex: passer un timeout en paramètre à un service lors de son appel)

Critères de succès d'une architecture microservice

Les services doivent être en ligne avec l'architecture d'affaires : l'organisation doit comprendre ce que le service fait pour supporter cette architecture

Les services doivent être faciles d'utilisation pour les consommateurs

Les services doivent permettre de s'adapter aux futurs besoins de l'entreprise : nouvelles capacités, nouveaux canaux de distribution, élasticité de la demande

Les services doivent être surveillés et sécurisés adéquatement

Une gouvernance doit être mis en place (structure, règles, guides, meilleures pratiques, etc.)