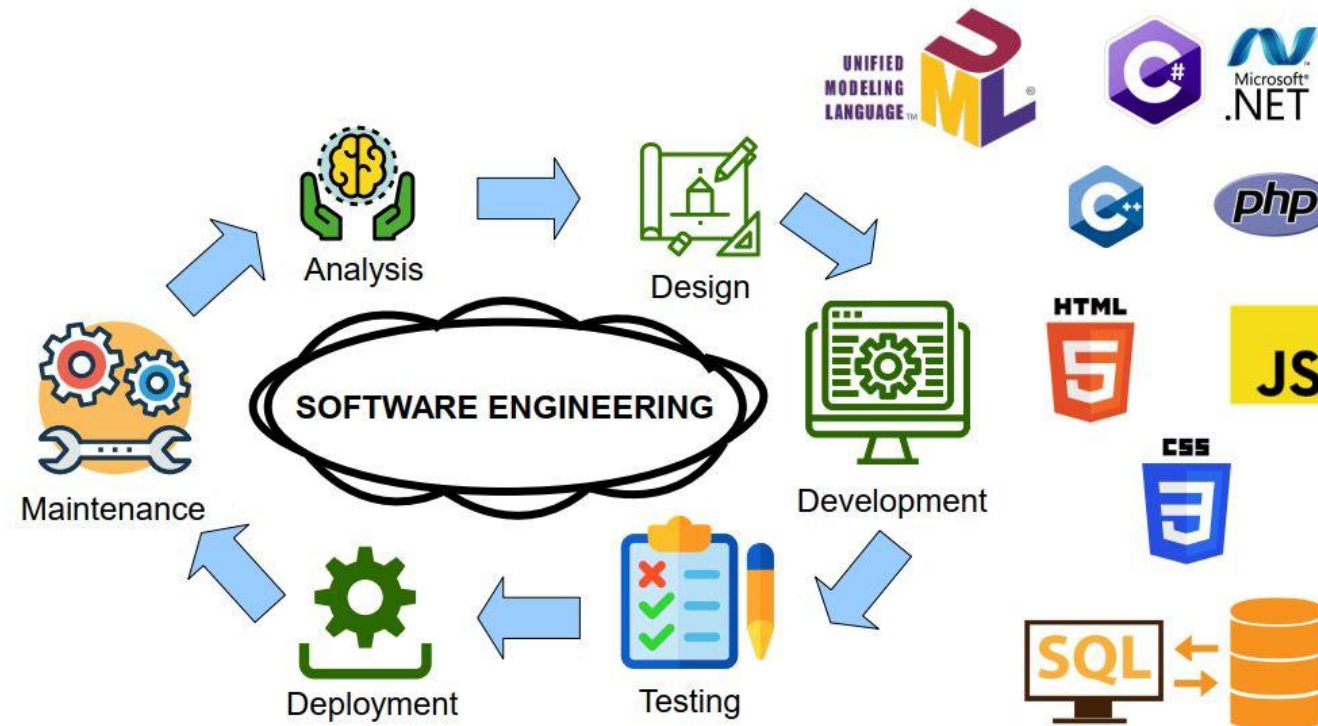


Architecture Logicielle



Dr El Hadji Bassirou TOURE
Université Cheikh Anta Diop de Dakar
2024 - 2025

Logiciels en cloud

Le cloud

- Le cloud est constitué d'un très grand nombre de serveurs distants proposés à la location par des entreprises propriétaires de ces serveurs.
 - Les serveurs en cloud sont des "serveurs virtuels", ce qui signifie qu'ils sont implémentés dans un logiciel plutôt que dans du matériel.
- Vous pouvez louer autant de serveurs que nécessaire, exécuter votre logiciel sur ces serveurs et les mettre à la disposition de vos clients.
 - Vos clients peuvent accéder à ces serveurs à partir de leurs propres ordinateurs ou d'autres appareils en réseau tels qu'une tablette ou un téléviseur.
 - Les serveurs en cloud peuvent être mis en service et arrêtés en fonction de l'évolution de la demande.
- Vous pouvez louer un serveur et installer vos propres logiciels ou payer pour accéder à des logiciels disponibles dans le cloud.

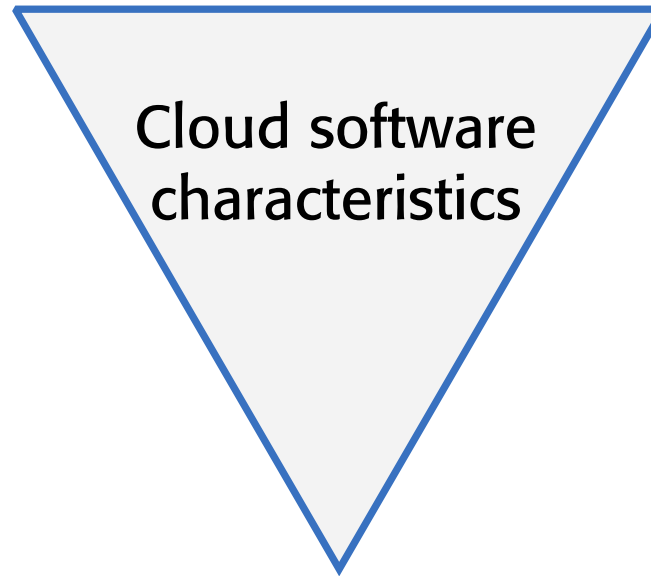
Évolutivité, élasticité et résilience

Scaleability

Maintain performance as
load increases

Elasticity

Adapt the server configuration
to changing demands



Resilience

Maintain service in the
event of server failure

Évolutivité, élasticité et résilience

- L'évolutivité reflète la capacité de votre logiciel à faire face à un nombre croissant d'utilisateurs.
 - Au fur et à mesure que la charge sur votre logiciel augmente, votre logiciel s'adapte automatiquement afin de maintenir les performances et le temps de réponse du système.
- L'élasticité est liée à l'évolutivité, mais elle permet également de réduire ou d'augmenter l'échelle.
 - En d'autres termes, vous pouvez surveiller la demande de votre application et ajouter ou supprimer des serveurs de manière dynamique en fonction de l'évolution du nombre d'utilisateurs.
- La résilience signifie que vous pouvez concevoir votre architecture logicielle de manière à tolérer les pannes de serveur.
 - Vous pouvez mettre à disposition plusieurs copies de votre logiciel simultanément. Si l'une d'entre elles tombe en panne, les autres continuent à fournir un service.

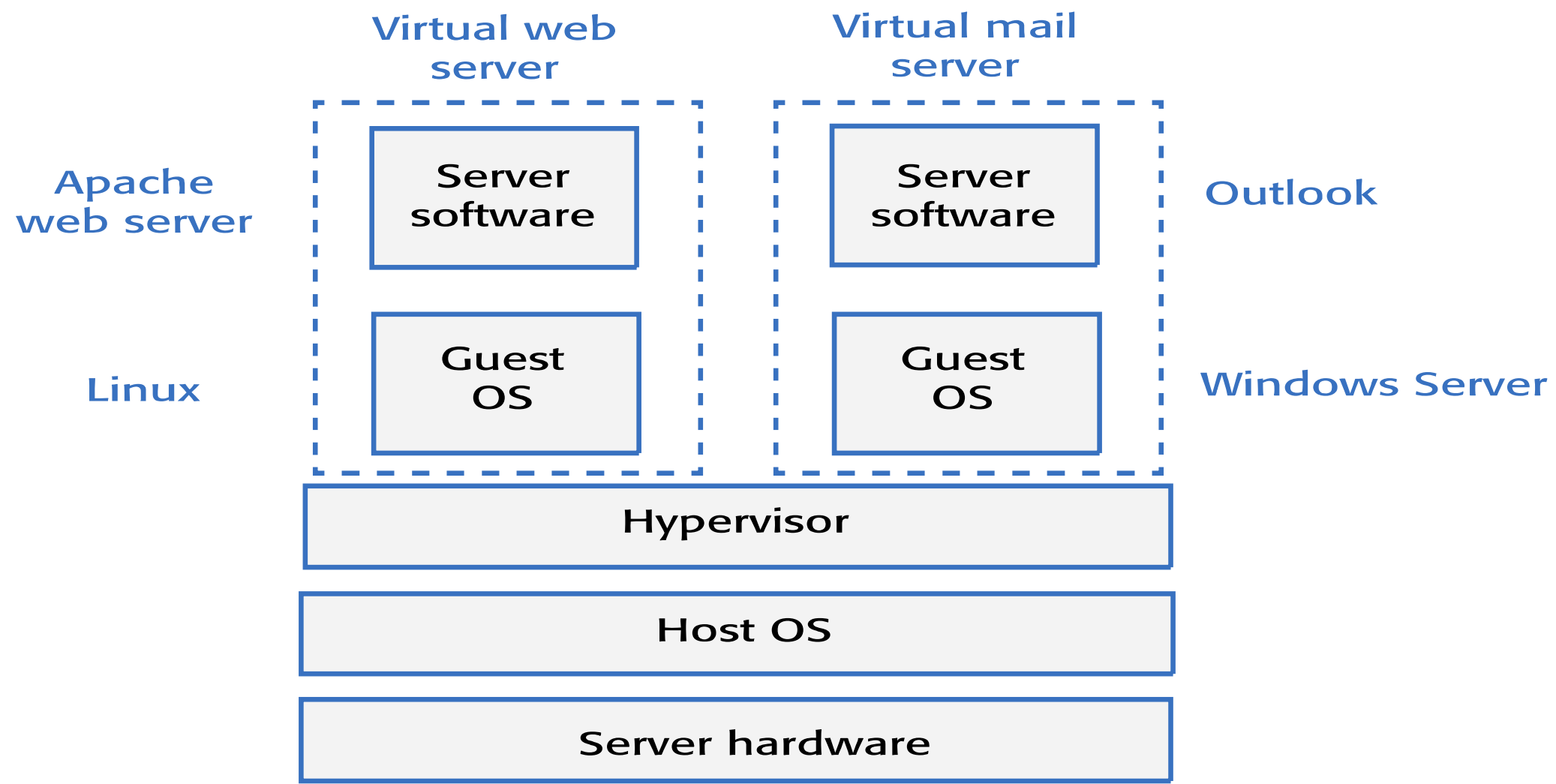
Avantages de l'utilisation du cloud pour le développement de logiciels

- Coût Vous évitez les coûts d'investissement initiaux liés à l'acquisition de matériel.
- Temps de démarrage Vous ne devez pas attendre la livraison du matériel pour commencer à travailler. Grâce au cloud, vous pouvez disposer de serveurs opérationnels en quelques minutes.
- Choix du serveur Si vous estimez que les serveurs que vous louez ne sont pas assez puissants, vous pouvez passer à des systèmes plus puissants. Vous pouvez ajouter des serveurs pour des besoins à court terme, tels que des tests de charge.
- Développement distribué Si vous avez une équipe de développement distribuée, travaillant sur différents sites, tous les membres de l'équipe disposent du même environnement de développement et peuvent partager toutes les informations de manière transparente.

Serveurs virtuels en cloud

- Un serveur virtuel fonctionne sur un ordinateur physique sous-jacent et se compose d'un système d'exploitation et d'un ensemble de logiciels qui fournissent les fonctionnalités requises pour le serveur.
 - Cette caractéristique d'"exécution n'importe où" est possible parce que le serveur virtuel n'a pas de dépendances externes
- Un serveur virtuel est un système autonome qui peut fonctionner sur n'importe quel matériel dans le cloud.
- Des machines virtuelles (VM), fonctionnant sur du matériel de serveur physique, peuvent être utilisées pour mettre en œuvre des serveurs virtuels.
 - Un hyperviseur fournit une émulation matérielle qui simule le fonctionnement du matériel sous-jacent.
- Si vous utilisez une machine virtuelle pour mettre en œuvre des serveurs virtuels, vous disposez exactement de la même plate-forme matérielle qu'un serveur physique.

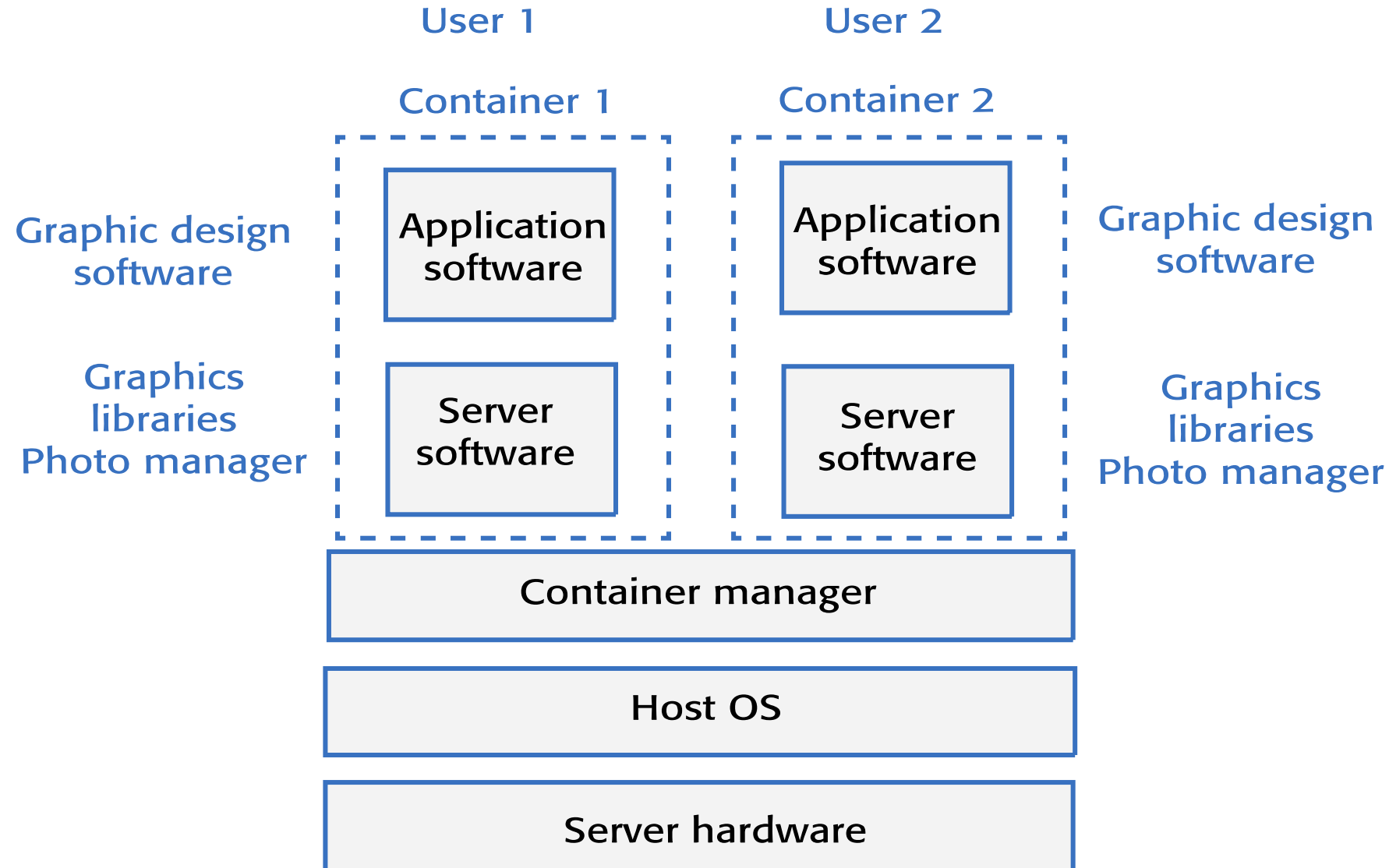
Mise en œuvre d'un serveur virtuel en tant que machine virtuelle



La virtualisation basée sur les conteneurs

- Si vous exploitez un système basé sur le cloud avec de nombreuses instances d'applications ou de services, qui utilisent toutes le même système d'exploitation, vous pouvez utiliser une technologie de virtualisation plus simple appelée "conteneurs".
- L'utilisation de conteneurs accélère le processus de déploiement de serveurs virtuels sur le cloud.
 - Les conteneurs ont généralement une taille de quelques mégaoctets, alors que les machines virtuelles ont une taille de quelques gigaoctets.
 - Les conteneurs peuvent être démarrés et arrêtés en quelques secondes au lieu des quelques minutes nécessaires pour une VM.
- Les conteneurs sont une technologie de virtualisation du système d'exploitation qui permet à des serveurs indépendants de partager un seul système d'exploitation.
 - Ils sont particulièrement utiles pour fournir des services d'application isolés où chaque utilisateur voit sa propre version d'une application.

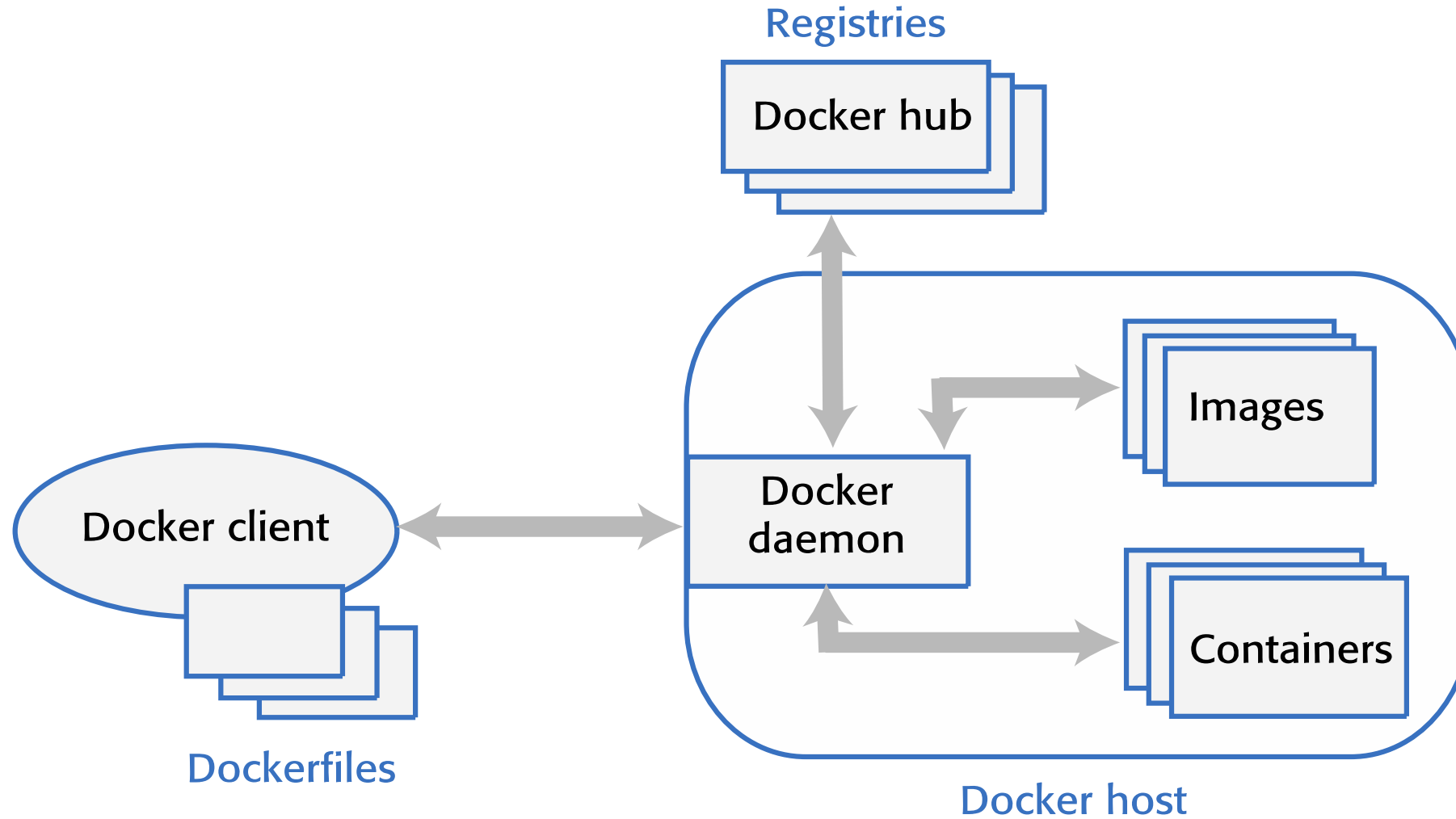
Utilisation de conteneurs pour fournir des services isolés



Docker

- Les conteneurs ont été développés par Google vers 2007, mais ils sont devenus une technologie courante vers 2015.
- Un projet open-source appelé Docker a fourni un moyen standard de gestion des conteneurs qui est rapide et facile à utiliser.
- Docker est un système de gestion de conteneurs qui permet aux utilisateurs de définir le logiciel à inclure dans un conteneur sous la forme d'une image Docker.
- Il comprend également un système d'exécution qui peut créer et gérer des conteneurs à l'aide de ces images Docker.

Le système de conteneurs Docker



Les éléments du système de conteneurs Docker

- Démon Docker Il s'agit d'un processus qui s'exécute sur un serveur hôte et qui est utilisé pour configurer, démarrer, arrêter et surveiller les conteneurs, ainsi que pour construire et gérer des images locales.
- Client Docker Ce logiciel est utilisé par les développeurs et les gestionnaires de systèmes pour définir et contrôler les conteneurs.
- Dockerfiles Les Dockerfiles définissent des applications exécutables (images) sous la forme d'une série de commandes d'installation qui spécifient le logiciel à inclure dans un conteneur. Chaque conteneur doit être défini par un Dockerfile associé.
- Image Un fichier Docker est interprété pour créer une image Docker, qui est un ensemble de répertoires avec les logiciels et les données spécifiés installés aux bons endroits. Les images sont configurées pour être des applications Docker exécutables.

Éléments du système de conteneurs Docker

- Hub Docker Il s'agit d'un registre d'images qui a été créé. Celles-ci peuvent être réutilisées pour configurer des conteneurs ou comme point de départ pour définir de nouvelles images.
- Conteneurs Les conteneurs sont des images d'exécution. Une image est chargée dans un conteneur et l'application définie par l'image commence à s'exécuter. Les conteneurs peuvent être déplacés d'un serveur à l'autre sans modification et répliqués sur plusieurs serveurs. Vous pouvez apporter des modifications à un conteneur Docker (par exemple en modifiant des fichiers), mais vous devez ensuite valider ces modifications pour créer une nouvelle image et redémarrer le conteneur.

Images Docker

- Les images Docker sont des répertoires qui peuvent être archivés, partagés et exécutés sur différents hôtes Docker. Tout ce qui est nécessaire pour faire fonctionner un système logiciel - binaires, bibliothèques, outils système, etc. est inclus dans le répertoire.
- Une image Docker est une couche de base, généralement issue du registre Docker, à laquelle sont ajoutés vos propres logiciels et données.
 - Le modèle en couches signifie que la mise à jour des applications Docker est rapide et efficace. Chaque mise à jour du système de fichiers est une couche au-dessus du système existant.
 - Pour modifier une application, il suffit d'envoyer les changements que vous avez apportés à son image, souvent un petit nombre de fichiers.

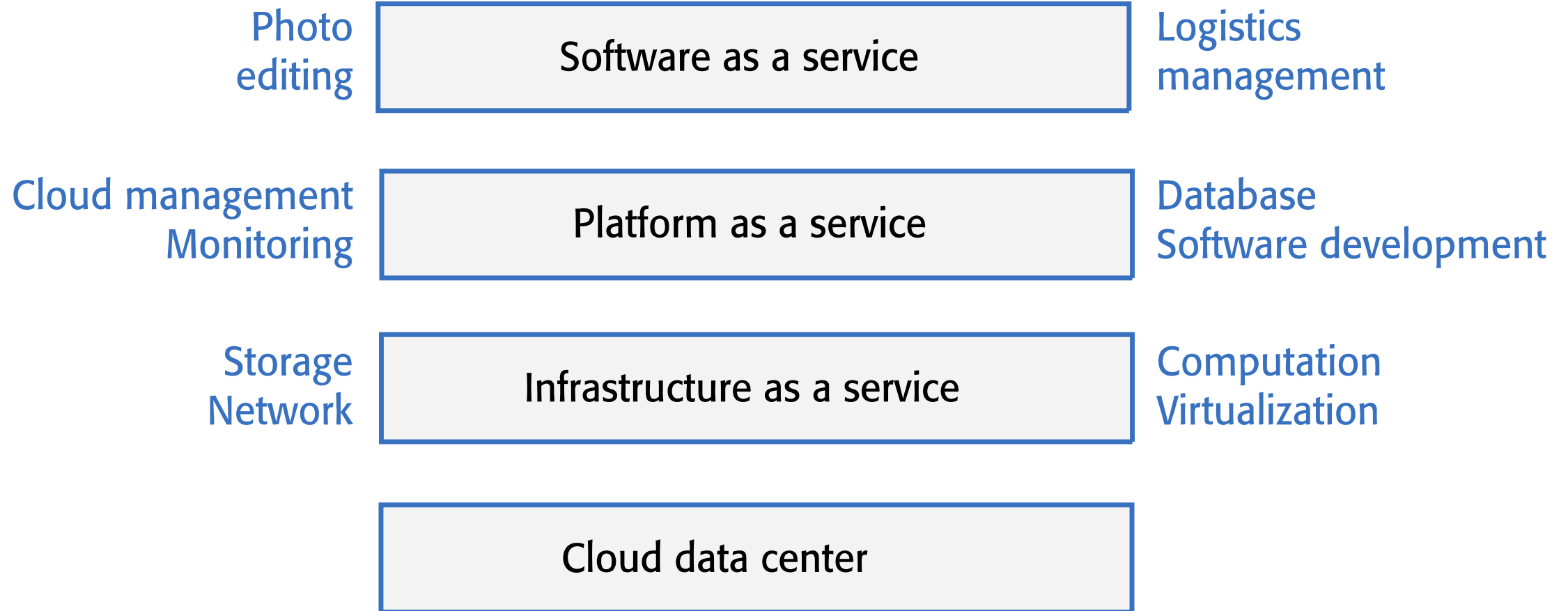
Avantages des conteneurs

- Ils résolvent le problème des dépendances logicielles. Vous n'avez pas à vous soucier du fait que les bibliothèques et autres logiciels du serveur d'application soient différents de ceux de votre serveur de développement.
 - Au lieu de livrer votre produit en tant que logiciel autonome, vous pouvez livrer un conteneur qui inclut tous les logiciels d'assistance dont votre produit a besoin.
- Ils fournissent un mécanisme de portabilité des logiciels entre différents clouds. Les conteneurs Docker peuvent fonctionner sur n'importe quel système ou fournisseur de cloud où le démon Docker est disponible.
- Ils fournissent un mécanisme efficace pour la mise en œuvre de services logiciels et soutiennent ainsi le développement d'architectures orientées services.
- Ils simplifient l'adoption de DevOps. Il s'agit d'une approche du support logiciel où la même équipe est responsable à la fois du développement et du support des logiciels opérationnels.

Tout en tant que service

- L'idée d'un service loué plutôt que possédé est fondamentale pour l'informatique en cloud.
- Infrastructure en tant que service (IaaS)
 - Les fournisseurs d'informatique en cloud proposent différents types de services d'infrastructure tels qu'un service de calcul, un service de réseau et un service de stockage que vous pouvez utiliser pour mettre en œuvre des serveurs virtuels.
- Plate-forme en tant que service (PaaS)
 - Il s'agit d'un niveau intermédiaire dans lequel vous utilisez des bibliothèques et des cadres fournis par le fournisseur de services en cloud pour mettre en œuvre votre logiciel. Ceux-ci permettent d'accéder à une série de fonctions, y compris les bases de données SQL et NoSQL.
- Logiciel en tant que service (SaaS)
 - Votre logiciel fonctionne dans le cloud et les utilisateurs y accèdent par l'intermédiaire d'un navigateur web ou d'une application mobile.

Tout en tant que service



Responsabilités de gestion pour IaaS et PaaS

Managed by
software provider

Software as a service

Managed by
software provider

Managed by
software provider

Application services
(database etc.)

Application services
(database etc.)

Managed by
cloud vendor

Managed by
software provider

Cloud management
services

Cloud management
services

Managed by
cloud vendor

Managed by
cloud vendor

Basic computational
services

Basic computational
services

Managed by
cloud vendor

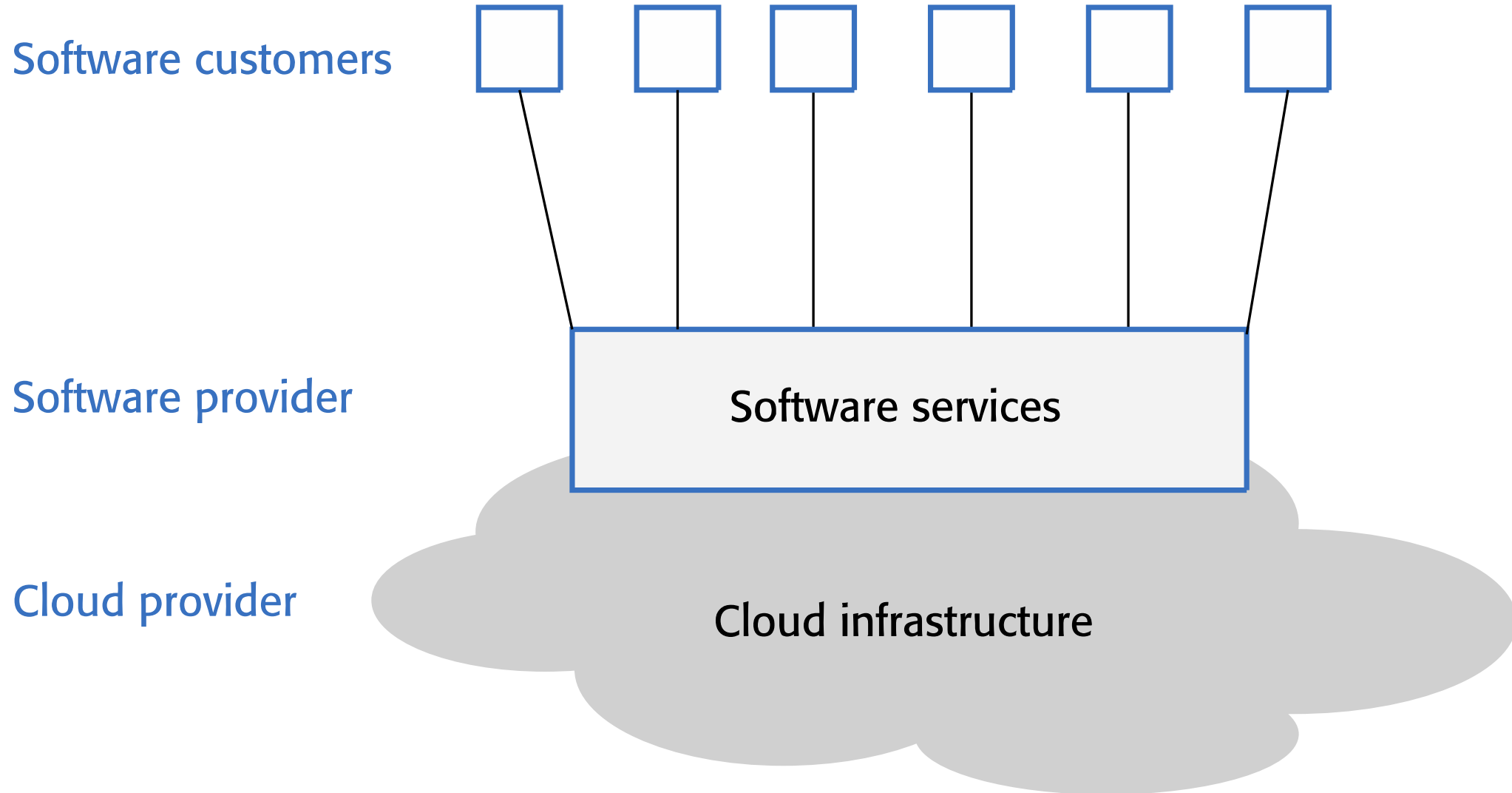
Infrastructure as a service

Platform as a service

Logiciel en tant que service

- De plus en plus, les produits logiciels sont fournis sous forme de service, plutôt qu'installés sur les ordinateurs des acheteurs.
- Si vous fournissez votre produit logiciel en tant que service, vous exécutez le logiciel sur vos serveurs, que vous pouvez louer auprès d'un fournisseur de services en cloud.
- Les clients n'ont pas besoin d'installer de logiciel et ils accèdent au système à distance par le biais d'un navigateur web ou d'une application mobile dédiée.
- Le modèle de paiement des logiciels en tant que service est généralement un modèle d'abonnement.
 - Les utilisateurs paient une redevance mensuelle pour utiliser le logiciel plutôt que de l'acheter.

Logiciel en tant que service



Avantages du SaaS pour les fournisseurs de logiciels

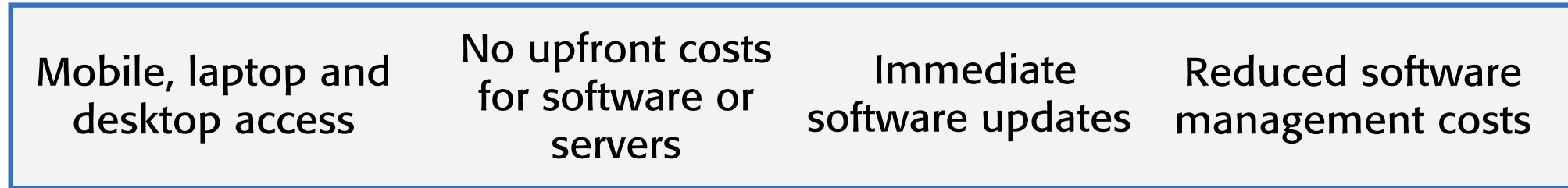
- Flux de trésorerie Les clients paient soit un abonnement régulier, soit au fur et à mesure qu'ils utilisent le logiciel. Cela signifie que vous disposez d'un flux de trésorerie régulier, avec des paiements tout au long de l'année. Vous ne vous trouvez pas dans une situation où vous bénéficiez d'une importante injection de liquidités lors de l'achat de produits, mais où vous ne percevez que très peu de revenus entre les sorties de produits.
- Gestion des mises à jour Vous contrôlez les mises à jour de votre produit et tous les clients reçoivent la mise à jour en même temps. Vous évitez le problème de l'utilisation et de la maintenance simultanées de plusieurs versions. Cela réduit vos coûts et facilite le maintien d'une base de code logicielle cohérente.
- Déploiement continu Vous pouvez déployer de nouvelles versions de votre logiciel dès que les modifications ont été apportées et testées. Cela signifie que vous pouvez corriger les bogues rapidement afin d'améliorer continuellement la fiabilité de votre logiciel

Avantages du SaaS pour les fournisseurs de logiciels

- Flexibilité de paiement Vous pouvez avoir plusieurs options de paiement différentes afin d'attirer un plus grand nombre de clients. Les petites entreprises ou les particuliers ne doivent pas être découragés par le coût initial élevé du logiciel.
- Essayez avant d'acheter Vous pouvez mettre rapidement à disposition des versions préliminaires gratuites ou à faible coût du logiciel afin de recueillir les commentaires des clients sur les bogues et sur la manière dont le produit pourrait être approuvé.
- Collecte de données Vous pouvez facilement collecter des données sur la manière dont le produit est utilisé et identifier ainsi les domaines à améliorer. Vous pouvez également recueillir des données sur les clients qui vous permettront de commercialiser d'autres produits auprès d'eux.

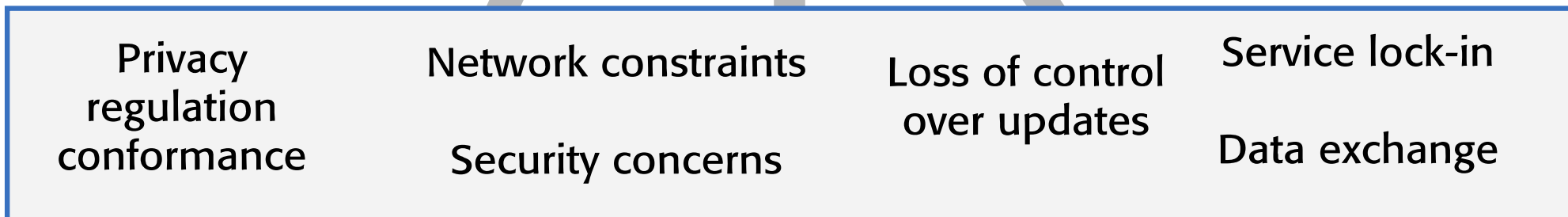
Avantages et inconvénients du SaaS pour les clients

Advantages



Software customer

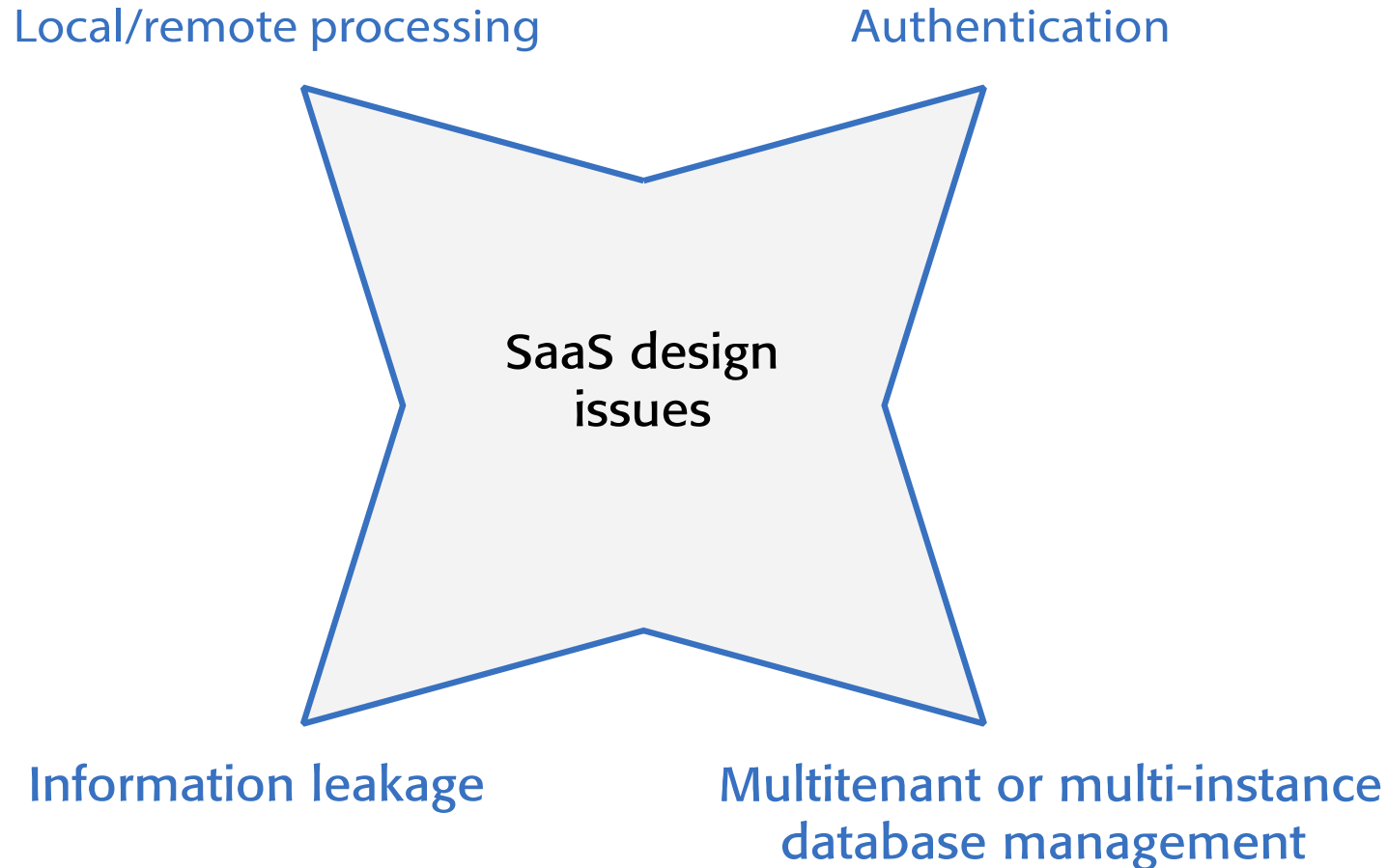
Disadvantages



Problèmes de stockage et de gestion des données pour le SaaS

- **Réglementation** Certains pays, comme les pays de l'UE, ont des lois strictes sur le stockage des informations personnelles. Ces lois peuvent être incompatibles avec les lois et réglementations du pays dans lequel le fournisseur de SaaS est basé. Si un fournisseur de SaaS ne peut pas garantir que ses lieux de stockage sont conformes aux lois du pays du client, les entreprises peuvent hésiter à utiliser son produit.
- **Transfert de données** Si l'utilisation d'un logiciel implique un important transfert de données, le temps de réponse du logiciel peut être limité par la vitesse du réseau. C'est un problème pour les particuliers et les petites entreprises qui n'ont pas les moyens de payer des connexions réseau à très haut débit.
- **Sécurité des données** Les entreprises qui traitent des informations sensibles peuvent être réticentes à confier le contrôle de leurs données à un fournisseur de logiciels externe. Comme nous l'avons vu dans un certain nombre d'affaires très médiatisées, même les grands fournisseurs de services en cloud ont connu des failles de sécurité. Vous ne pouvez pas supposer qu'ils offrent toujours une meilleure sécurité que les propres serveurs du client.
- **Échange de données** Si vous devez échanger des données entre un service en cloud et d'autres services ou applications logicielles locales, cela peut s'avérer difficile, à moins que le service en cloud ne fournisse une API accessible pour un usage externe

Questions relatives à la conception d'un logiciel fourni en tant que service



Questions relatives à la conception de SaaS (1)

- Traitement local/à distance

- Un produit logiciel peut être conçu de manière à ce que certaines fonctionnalités soient exécutées localement dans le navigateur ou l'application mobile de l'utilisateur et d'autres sur un serveur distant.
- L'exécution locale réduit le trafic sur le réseau et augmente donc la vitesse de réponse de l'utilisateur. Cela est utile lorsque les utilisateurs disposent d'une connexion réseau lente.
- Le traitement local augmente la puissance électrique nécessaire pour faire fonctionner le système.

- L'authentification

- Si vous mettez en place votre propre système d'authentification, les utilisateurs doivent se souvenir d'une autre série d'informations d'identification.
- De nombreux systèmes permettent l'authentification à l'aide des identifiants Google, Facebook ou LinkedIn de l'utilisateur.
- Pour les produits professionnels, vous devrez peut-être mettre en place un système d'authentification fédérée, qui délègue l'authentification à l'entreprise dans laquelle l'utilisateur travaille.

Questions relatives à la conception de SaaS (2)

- Fuite d'informations
 - Si vous avez plusieurs utilisateurs appartenant à plusieurs organisations, la fuite d'informations d'une organisation à l'autre constitue un risque pour la sécurité.
 - Cela peut se produire de différentes manières, c'est pourquoi vous devez être très prudent lors de la conception de votre système de sécurité afin d'éviter cela.
- Systèmes multi-tenants et multi-instances
 - Dans un système multi-tenants, tous les clients sont servis par une instance unique du système et une base de données multi-tenants.
- Dans un système multi-instances, une copie distincte du système et de la base de données est mise à la disposition de chaque utilisateur.

Systèmes multi-tenants

- Une base de données multi-tenants est partitionnée de manière à ce que les entreprises clientes disposent de leur propre espace et puissent stocker et accéder à leurs propres données.
 - Il existe un schéma de base de données unique, défini par le fournisseur SaaS, qui est partagé par tous les utilisateurs du système.
 - Les éléments de la base de données sont marqués d'un identifiant de tenant, représentant une entreprise qui a stocké des données dans le système. Le logiciel d'accès à la base de données utilise cet identifiant de tenant pour fournir une "isolation logique", ce qui signifie que les utilisateurs semblent travailler avec leur propre base de données.

Exemple de base de données multi-tenants

| Stock management | | | | | |
|------------------|-----|---------|-------|----------|-----------|
| Tenant | Key | Item | Stock | Supplier | Ordered |
| T516 | 100 | Widg 1 | 27 | S13 | 2017/2/12 |
| T632 | 100 | Obj 1 | 5 | S13 | 2017/1/11 |
| T973 | 100 | Thing 1 | 241 | S13 | 2017/2/7 |
| T516 | 110 | Widg 2 | 14 | S13 | 2017/2/2 |
| T516 | 120 | Widg 3 | 17 | S13 | 2017/1/24 |
| T973 | 100 | Thing 2 | 132 | S26 | 2017/2/12 |

Avantages des bases de données multi-tenants

- **Utilisation des ressources** Le fournisseur de SaaS contrôle toutes les ressources utilisées par le logiciel et peut optimiser le logiciel pour une utilisation efficace de ces ressources.
- **Sécurité** Les bases de données multitenants doivent être conçues pour la sécurité, car les données de tous les clients sont conservées dans la même base de données. Elles sont donc susceptibles de présenter moins de failles de sécurité que les bases de données standard. La gestion de la sécurité est simplifiée car il n'y a qu'une seule copie du logiciel de la base de données à corriger en cas de découverte d'une faille de sécurité.
- **Gestion des mises à jour** Il est plus facile de mettre à jour une seule instance de logiciel que plusieurs instances. Les mises à jour sont fournies à tous les clients en même temps, de sorte que tous utilisent la dernière version du logiciel.

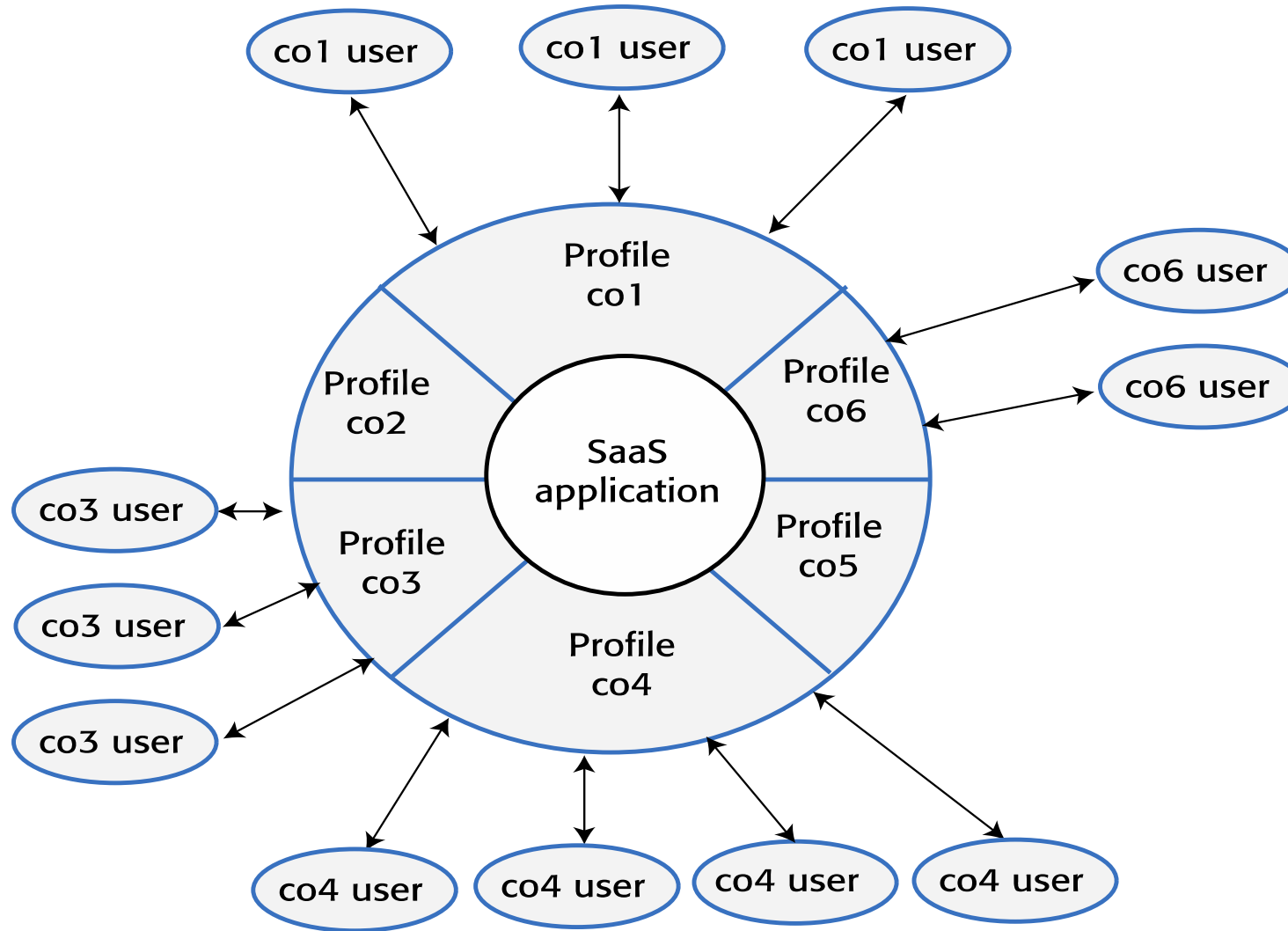
Inconvénients des bases de données multi-tenants

- **Inflexibilité** Les clients doivent tous utiliser le même schéma de base de données, avec une marge de manœuvre limitée pour adapter ce schéma aux besoins individuels. Les adaptations possibles de la base de données sont expliquées plus loin dans cette section.
- **Sécurité** Les données de tous les clients étant conservées dans la même base de données, il existe une possibilité théorique de fuite de données d'un client à l'autre. En fait, il y a très peu de cas où cela se produit. Plus grave encore, une faille dans la sécurité de la base de données affecte tous les clients.
- **Complexité** Les systèmes multitenants sont généralement plus complexes que les systèmes multi-instances en raison de la nécessité de gérer de nombreux utilisateurs. Il y a donc une probabilité accrue de bogues dans le logiciel de la base de données.

Personnalisations possibles pour SaaS

- **Authentification** Les entreprises peuvent souhaiter que les utilisateurs s'authentifient à l'aide de leurs identifiants professionnels plutôt qu'à l'aide des identifiants de compte définis par le fournisseur de logiciels. Au chapitre 7, j'explique comment l'authentification fédérée rend cela possible.
- **Image de marque** Les entreprises peuvent vouloir une interface utilisateur qui soit à l'image de leur propre organisation.
- **Règles de gestion** Les entreprises peuvent vouloir définir leurs propres règles de gestion et flux de travail applicables à leurs propres données.
- **Schémas de données** Les entreprises peuvent vouloir étendre le modèle de données standard utilisé dans la base de données du système pour répondre à leurs propres besoins.
- **Contrôle d'accès** Les entreprises peuvent vouloir définir leur propre modèle de contrôle d'accès qui détermine les données auxquelles des utilisateurs ou groupes d'utilisateurs spécifiques peuvent accéder et les opérations autorisées sur ces données.

Profils d'utilisateurs pour l'accès SaaS



Extensibilité de la base de données à l'aide de champs supplémentaires

Figure 5.12 Database extensibility using additional fields

| Stock management | | | | | | | | |
|------------------|-----|---------|-------|----------|-----------|-------|-------|-------|
| Tenant | Key | Item | Stock | Supplier | Ordered | Ext 1 | Ext 2 | Ext 3 |
| T516 | 100 | Widg 1 | 27 | S13 | 2017/2/12 | | | |
| T632 | 100 | Obj 1 | 5 | S13 | 2017/1/11 | | | |
| T973 | 100 | Thing 1 | 241 | S13 | 2017/2/7 | | | |
| T516 | 110 | Widg 2 | 14 | S13 | 2017/2/2 | | | |
| T516 | 120 | Widg 3 | 17 | S13 | 2017/1/24 | | | |
| T973 | 100 | Thing 2 | 132 | S26 | 2017/2/12 | | | |

Ajouter des champs pour étendre la base de données

- Vous ajoutez des colonnes supplémentaires à chaque table de la base de données et définissez un profil de client qui associe les noms de colonnes souhaités par le client à ces colonnes supplémentaires. Cependant, vous ne pouvez pas vous contenter d'ajouter des colonnes supplémentaires à chaque table de la base de données :
 - Il est difficile de savoir combien de colonnes supplémentaires vous devez inclure. Si vous en avez trop peu, les clients trouveront qu'il n'y en a pas assez pour ce qu'ils ont à faire.
 - Si vous vous adressez à des clients qui ont besoin d'un grand nombre de colonnes supplémentaires, vous constaterez que la plupart d'entre eux ne les utilisent pas, ce qui entraînera une perte d'espace dans votre base de données.
 - Différents clients sont susceptibles d'avoir besoin de différents types de colonnes.
 - Par exemple, certains clients peuvent souhaiter avoir des colonnes dont les éléments sont des chaînes de caractères, d'autres peuvent souhaiter avoir des colonnes qui sont des nombres entiers.
 - Vous pouvez contourner ce problème en gérant tous les éléments comme des chaînes de caractères. Toutefois, cela signifie que vous ou votre client devez fournir un logiciel de conversion pour créer des éléments du bon type.

Extension d'une base de données à l'aide de tables

- Une autre approche de l'extensibilité des bases de données consiste à permettre aux clients d'ajouter un nombre quelconque de champs supplémentaires et de définir les noms, les types et les valeurs de ces champs.
- Les noms et les types de ces valeurs sont conservés dans une table distincte, à laquelle on accède à l'aide de l'identifiant du tenant.
- Malheureusement, l'utilisation de tables de cette manière ajoute de la complexité au logiciel de gestion de base de données.
 - Des tables supplémentaires doivent être gérées et les informations qu'elles contiennent doivent être intégrées dans la base de données.

Extensibilité de la base de données à l'aide de tables

Main database table

Tab1

| Stock management | | | | | | |
|------------------|-----|---------|-------|----------|-----------|-------|
| Tenant | ID | Item | Stock | Supplier | Ordered | Ext 1 |
| T516 | 100 | Widg 1 | 27 | S13 | 2017/2/12 | E123 |
| T632 | 100 | Obj 1 | 5 | S13 | 2017/1/11 | E200 |
| T973 | 100 | Thing 1 | 241 | S13 | 2017/2/7 | E346 |
| T516 | 110 | Widg 2 | 14 | S13 | 2017/2/2 | E124 |
| T516 | 120 | Widg 3 | 17 | S13 | 2017/1/24 | E125 |
| T973 | 100 | Thing 2 | 132 | S26 | 2017/2/12 | E347 |

Tab2

| Field names | | |
|-------------|-------------|---------|
| Tenant | Name | Type |
| T516 | 'Location' | String |
| T516 | 'Weight' | Integer |
| T516 | 'Fragile' | Bool |
| T632 | 'Delivered' | Date |
| T632 | 'Place' | String |
| T973 | 'Delivered' | Date |

Extension table showing the field names for each company that needs database extensions

Tab3

| Field values | | |
|--------------|--------|-------------|
| Record | Tenant | Value |
| E123 | T516 | 'A17/S6' |
| E123 | T516 | '4' |
| E123 | T516 | 'False' |
| E200 | T632 | '2017/1/15' |
| E200 | T632 | 'Dublin' |
| E346 | T973 | '2017/2/10' |
| ... | | |

Value table showing the value of extension fields for each record

Sécurité des bases de données

- Les informations relatives à tous les clients sont stockées dans la même base de données dans un système multitenant, de sorte qu'un bogue logiciel ou une attaque pourrait exposer les données d'une partie ou de la totalité des clients à d'autres personnes.
- Les principaux problèmes de sécurité sont le contrôle d'accès à plusieurs niveaux et le cryptage.
 - Le contrôle d'accès à plusieurs niveaux signifie que l'accès aux données doit être contrôlé à la fois au niveau de l'organisation et au niveau individuel.
 - Vous devez disposer d'un contrôle d'accès au niveau de l'organisation pour vous assurer que les opérations effectuées sur la base de données ne concernent que les données de cette organisation. L'utilisateur individuel qui accède aux données doit également disposer de ses propres autorisations d'accès.
- Le cryptage des données dans une base de données multitenant garantit aux utilisateurs de l'entreprise que leurs données ne peuvent pas être consultées par des personnes d'autres entreprises en cas de défaillance du système.

Bases de données multi-instances

- Les systèmes multi-instances sont des systèmes SaaS dans lesquels chaque client dispose de son propre système adapté à ses besoins, y compris sa propre base de données et ses propres contrôles de sécurité.
- Les systèmes multi-instances basés sur l'informatique en cloud sont conceptuellement plus simples que les systèmes multi-tenants et évitent les problèmes de sécurité tels que la fuite de données d'une organisation à l'autre.
- Il existe deux types de systèmes multi-instances :
 - Les systèmes multi-instances basés sur des machines virtuelles sont des systèmes multi-instances dans lesquels l'instance logicielle et la base de données de chaque client s'exécutent dans sa propre machine virtuelle. Tous les utilisateurs d'un même client peuvent accéder à la base de données partagée du système.
 - Systèmes multi-instances basés sur des conteneurs* Il s'agit de systèmes multi-instances dans lesquels chaque utilisateur dispose d'une version isolée du logiciel et de la base de données fonctionnant dans un ensemble de conteneurs.
 - Cette approche est adaptée aux produits dans lesquels les utilisateurs travaillent principalement de manière indépendante, avec relativement peu de partage de données. Par conséquent, elle convient mieux aux logiciels destinés aux particuliers qu'aux entreprises ou aux produits commerciaux qui ne sont pas très gourmands en données.

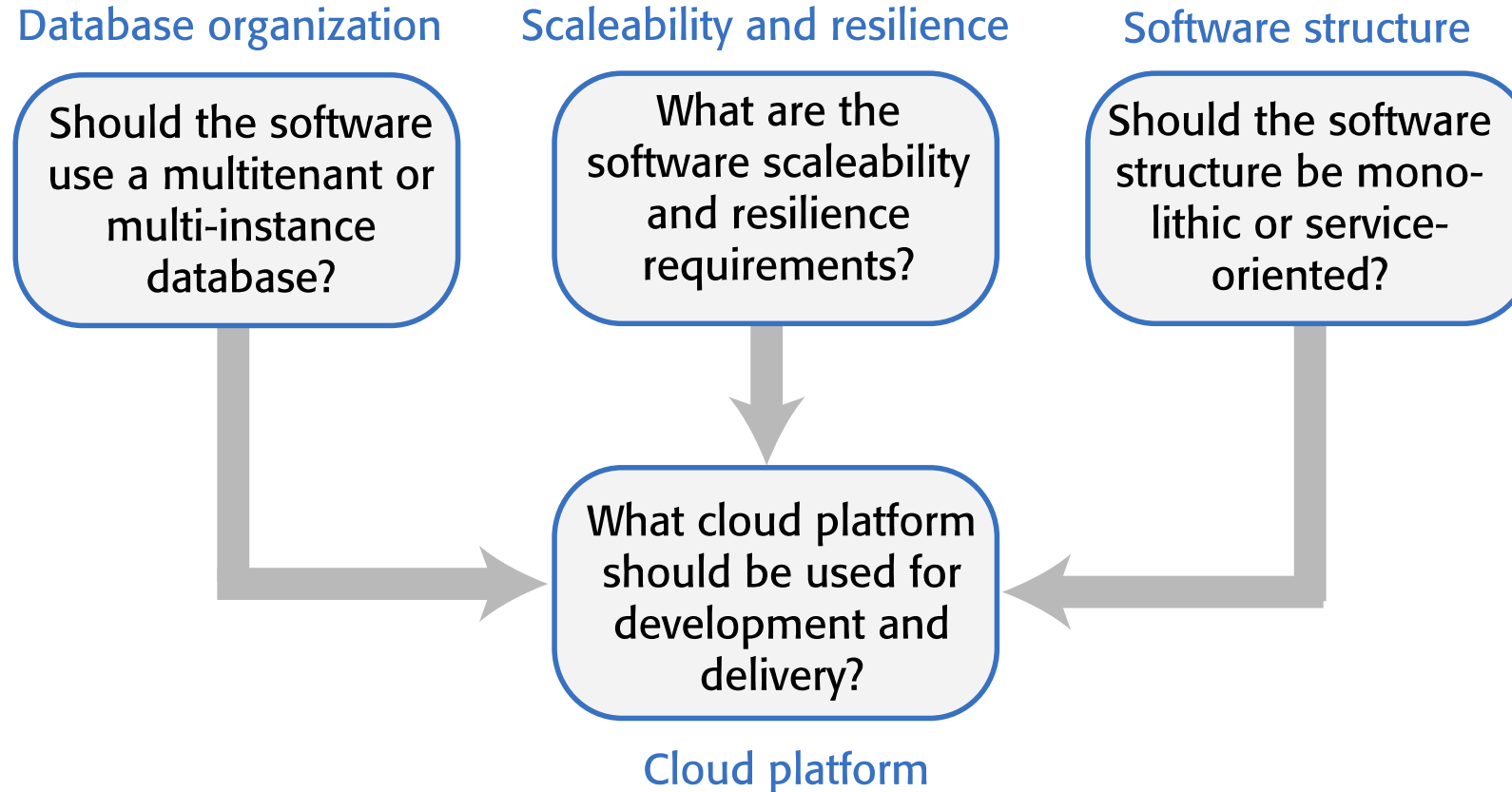
Avantages des bases de données multi-instances

- **Flexibilité** Chaque instance du logiciel peut être personnalisée et adaptée aux besoins d'un client. Les clients peuvent utiliser des schémas de base de données complètement différents et il est facile de transférer des données d'une base de données client vers la base de données produit.
- **Sécurité** Chaque client possède sa propre base de données, de sorte qu'il n'y a aucune possibilité de fuite de données d'un client à l'autre.
- **Évolutivité** Les instances du système peuvent être adaptées aux besoins de chaque client. Par exemple, certains clients peuvent avoir besoin de serveurs plus puissants que d'autres.
- **Résilience** Si une défaillance logicielle se produit, elle n'affectera probablement qu'un seul client. Les autres clients peuvent continuer à travailler normalement.

Inconvénients des bases de données multi-instances

- Coût L'utilisation de systèmes multi-instances est plus onéreuse en raison des coûts de location de nombreuses VM dans le cloud et des coûts de gestion de systèmes multiples. En raison de la lenteur du démarrage, il peut être nécessaire de louer des machines virtuelles et de les faire fonctionner en permanence, même si la demande pour le service est très faible.
- Gestion des mises à jour Il est plus complexe de gérer les mises à jour du logiciel car de nombreuses instances doivent être mises à jour. Cela est particulièrement problématique lorsque les instances individuelles ont été adaptées aux besoins de clients spécifiques.

Décisions architecturales pour l'ingénierie logicielle en cloud



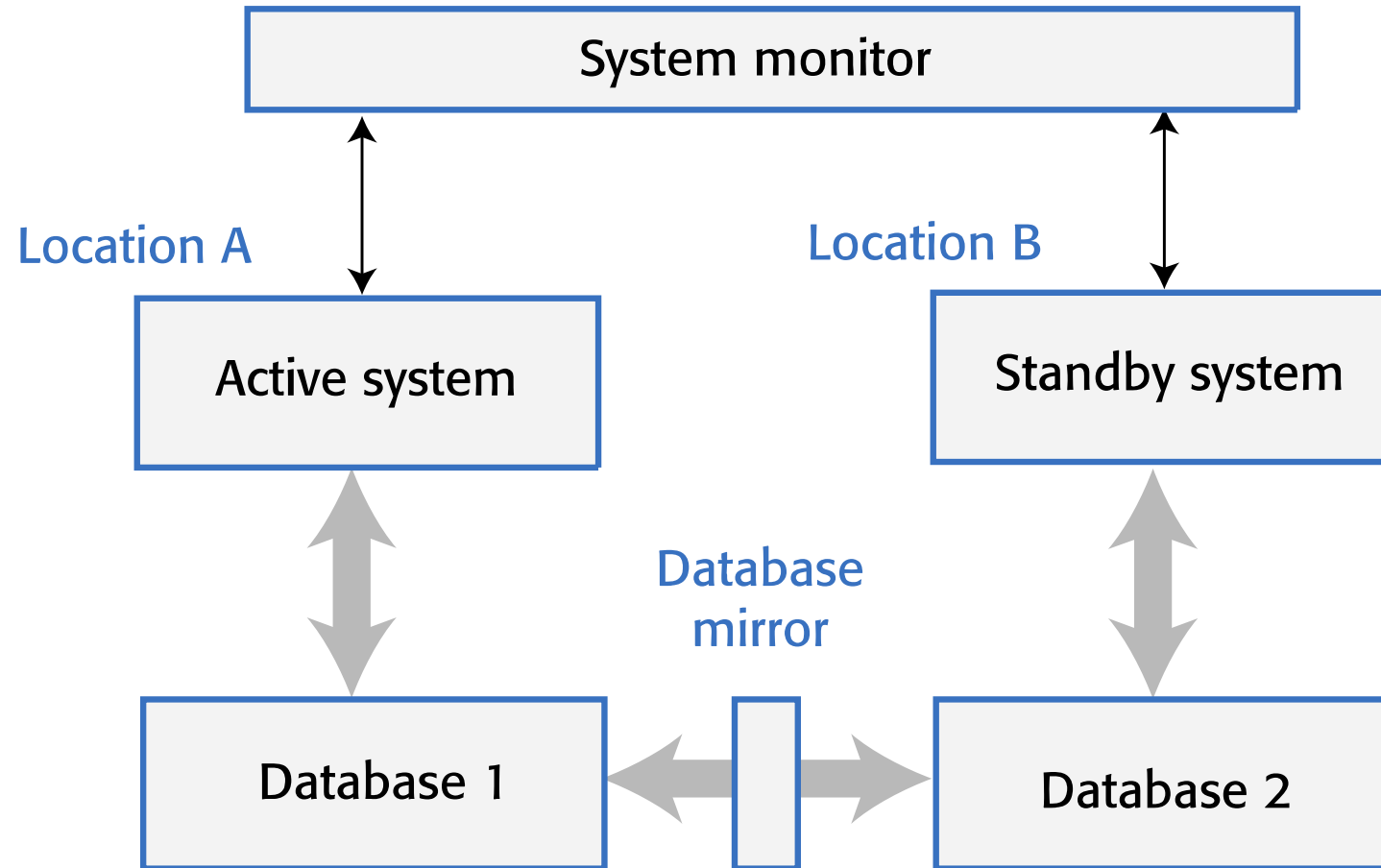
Questions à poser lors du choix d'une organisation de base de données

- Clients cibles Les clients ont-ils besoin de schémas de base de données différents et d'une personnalisation de la base de données ? Les clients ont-ils des préoccupations en matière de sécurité concernant le partage de la base de données ? Dans ce cas, utilisez une base de données multi-instances.
- Exigences en matière de transactions Vos produits doivent-ils impérativement prendre en charge les transactions ACID, qui garantissent la cohérence des données à tout moment ? Si c'est le cas, utilisez une base de données multi-tenants ou une base de données multi-instances basée sur une VM.
- Taille de la base de données et connectivité Quelle est la taille de la base de données type utilisée par les clients ? Combien y a-t-il de relations entre les éléments de la base de données ? Un modèle multi-tenants est généralement préférable pour les très grandes bases de données, car il permet de concentrer les efforts sur l'optimisation des performances.
- Interopérabilité des bases de données Les clients souhaiteront-ils transférer des informations à partir de bases de données existantes ? Quelles sont les différences de schémas entre ces bases et une éventuelle base de données multitenant ? Quelle assistance logicielle attendront-ils pour effectuer le transfert de données ? Si les clients ont de nombreux schémas différents, il convient d'utiliser une base de données multi-instances.
- Structure du système Votre système utilise-t-il une architecture orientée services ? Les bases de données clients peuvent-elles être divisées en un ensemble de bases de données de services individuels ? Si c'est le cas, utilisez des bases de données conteneurisées et multi-instances.

Évolutivité et résilience

- L'évolutivité d'un système reflète sa capacité à s'adapter automatiquement aux changements de charge sur ce système.
- La résilience d'un système reflète sa capacité à continuer à fournir des services essentiels en cas de défaillance ou d'utilisation malveillante du système.
- L'évolutivité d'un système se traduit par la possibilité d'ajouter de nouveaux serveurs virtuels (scaling-out) ou d'augmenter la puissance d'un serveur système (scaling-up) en réponse à l'augmentation de la charge.
 - Dans les systèmes basés sur l'informatique en cloud, la mise à l'échelle plutôt que l'augmentation est l'approche normale utilisée. Votre logiciel doit être organisé de manière à ce que les composants logiciels individuels puissent être répliqués et exécutés en parallèle.
- Pour atteindre la résilience, vous devez être en mesure de redémarrer votre logiciel rapidement après une défaillance matérielle ou logicielle.

Utilisation d'un système de secours pour assurer la résilience



La résilience

- La résilience repose sur la redondance :
 - Des répliques du logiciel et des données sont conservées à différents endroits.
 - Les mises à jour de la base de données sont mises en miroir de manière à ce que la base de données de secours soit une copie fonctionnelle de la base de données opérationnelle.
 - Un moniteur système vérifie en permanence l'état du système. Il peut basculer automatiquement vers le système de secours en cas de défaillance du système opérationnel.
- Vous devez utiliser des serveurs virtuels redondants qui ne sont pas hébergés sur le même ordinateur physique et placer les serveurs à différents endroits.
 - Idéalement, ces serveurs devraient être situés dans des centres de données différents.
 - En cas de défaillance d'un serveur physique ou d'un centre de données plus vaste, il est possible de basculer automatiquement vers les copies logicielles situées ailleurs.

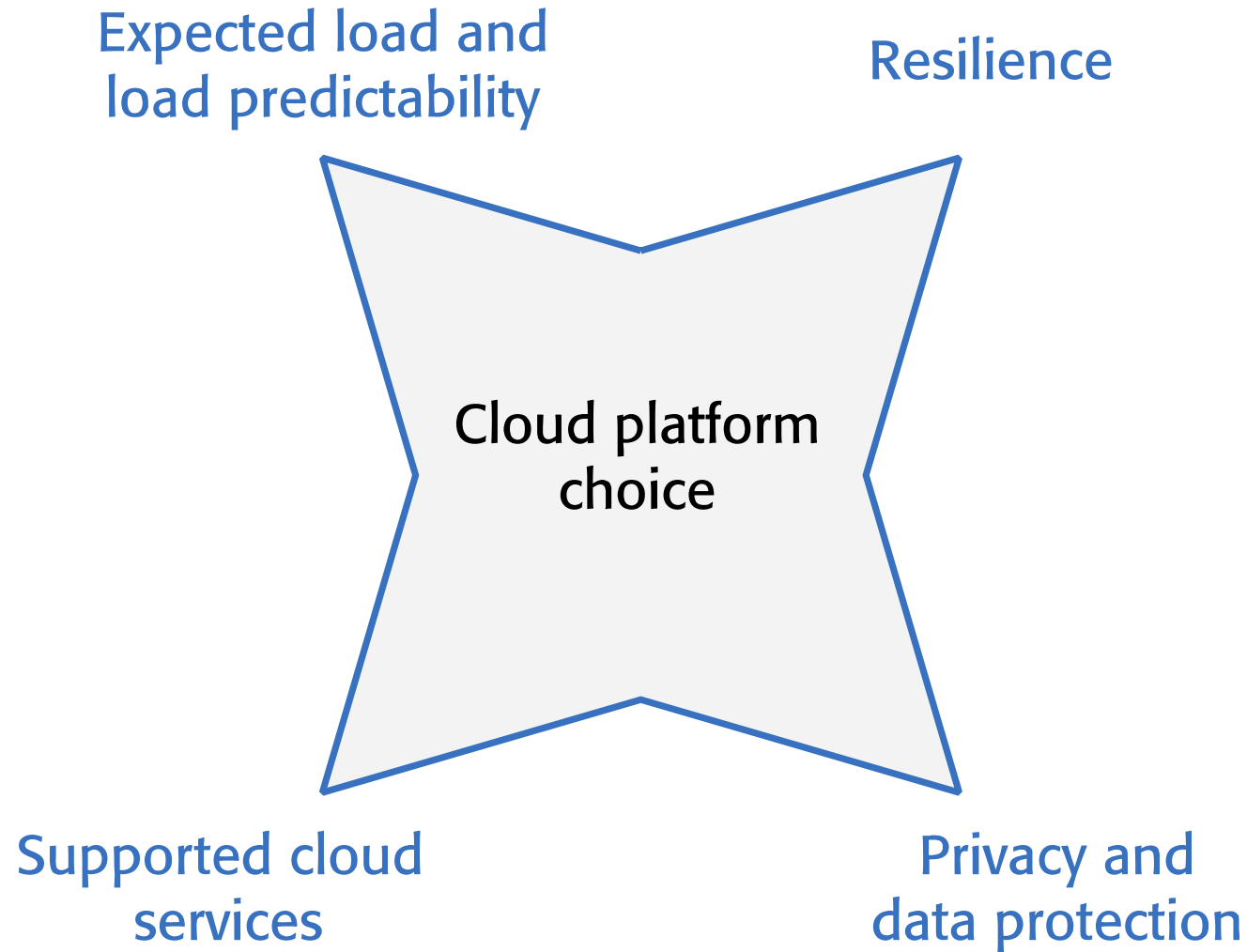
Structure du système

- Une approche orientée objet du génie logiciel a été largement utilisée pour le développement de systèmes client-serveur construits autour d'une base de données partagée.
- Le système lui-même est, logiquement, un système monolithique distribué sur plusieurs serveurs exécutant de grands composants logiciels. L'architecture client-serveur multi-niveaux traditionnelle est basée sur ce modèle de système distribué.
- L'alternative à l'approche monolithique de l'organisation logicielle est une approche orientée services, dans laquelle le système est décomposé en services à grain fin et sans état.
 - Parce qu'il est sans état, chaque service est indépendant et peut être répliqué, distribué et migré d'un serveur à l'autre.
 - L'approche orientée services est particulièrement adaptée aux logiciels en cloud, les services étant déployés dans des conteneurs.

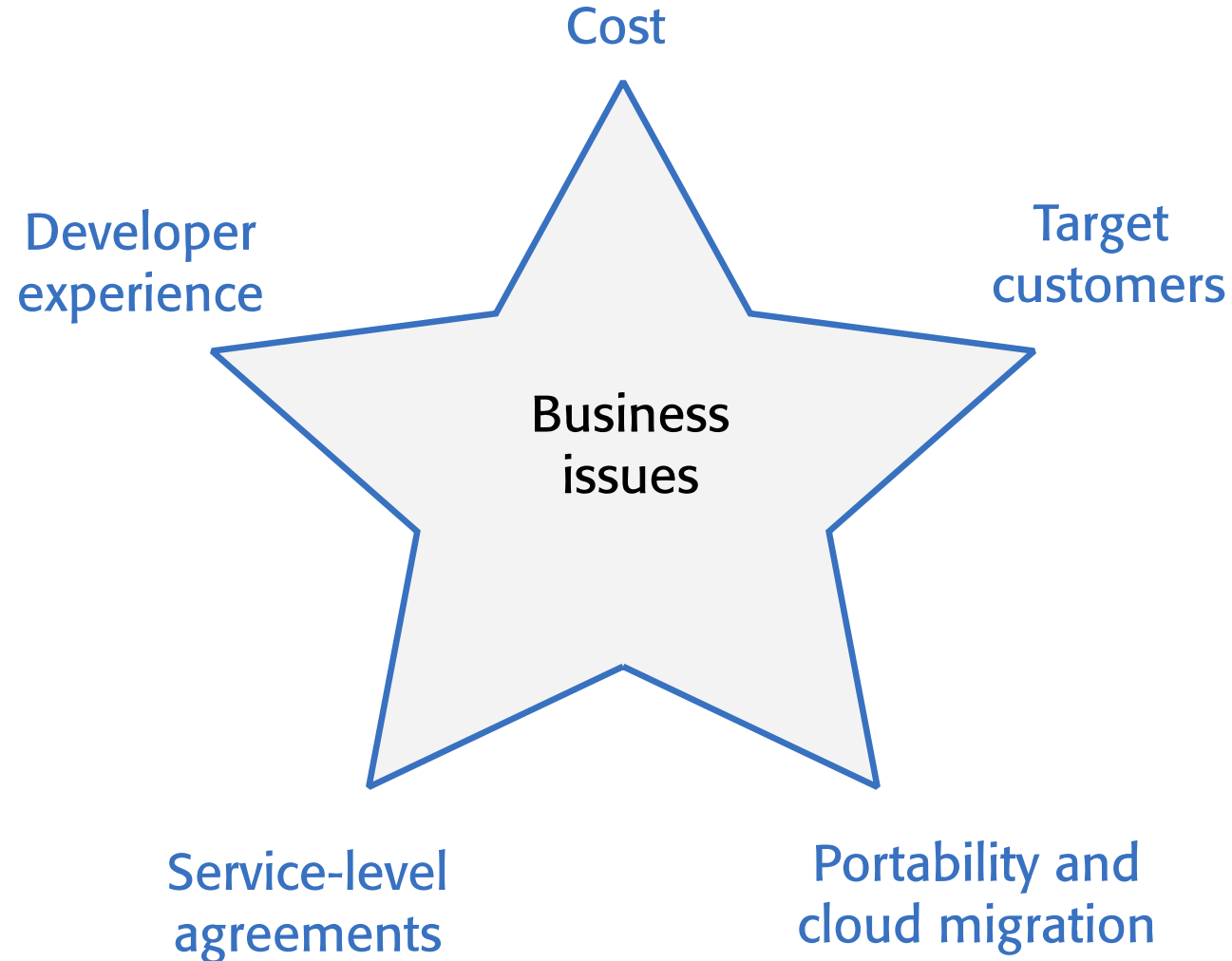
Plate-forme en cloud

- Les plates-formes en cloud comprennent des plates-formes à usage général comme Amazon Web Services ou des plates-formes moins connues axées sur une application spécifique, comme la SAP Cloud Platform. Il existe également des fournisseurs nationaux plus petits qui offrent des services plus limités, mais qui peuvent être plus enclins à adapter leurs services aux besoins des différents clients.
- Il n'existe pas de "meilleure" plateforme et vous devez choisir un fournisseur de services en cloud en fonction de votre formation et de votre expérience, du type de produit que vous développez et des attentes de vos clients.
- Vous devez tenir compte à la fois des aspects techniques et des aspects commerciaux lorsque vous choisissez une plateforme en cloud pour votre produit.

Problèmes techniques liés au choix d'une plate-forme en cloud



Questions commerciales liées au choix d'une plateforme en cloud



Points clés 1

- Le cloud est constitué d'un grand nombre de serveurs virtuels que vous pouvez louer pour votre propre usage. Vous et vos clients accédez à ces serveurs à distance via l'internet et payez pour le temps d'utilisation du serveur.
- La virtualisation est une technologie qui permet d'exécuter plusieurs instances de serveur sur le même ordinateur physique. Cela signifie que vous pouvez créer des instances isolées de votre logiciel pour le déployer dans le cloud.
- Les machines virtuelles sont des répliques de serveurs physiques sur lesquelles vous exécutez votre propre système d'exploitation, votre pile technologique et vos applications.
- Les conteneurs sont une technologie de virtualisation légère qui permet de répliquer et de déployer rapidement des serveurs virtuels. Tous les conteneurs utilisent le même système d'exploitation. Docker est actuellement la technologie de conteneur la plus utilisée.
- L'une des caractéristiques fondamentales de l'cloud est que "tout" peut être fourni en tant que service et accessible via l'internet. Un service est loué plutôt que possédé et est partagé avec d'autres utilisateurs.

Points clés 2

- L'infrastructure en tant que service (IaaS) signifie que l'informatique, le stockage et d'autres services sont disponibles dans le cloud. Il n'est pas nécessaire d'exploiter vos propres serveurs physiques.
- La plate-forme en tant que service (PaaS) consiste à utiliser des services fournis par un fournisseur de plate-forme en cloud pour permettre l'auto-évaluation de votre logiciel en réponse à la demande.
- Le logiciel en tant que service (SaaS) signifie que le logiciel d'application est fourni en tant que service aux utilisateurs. Cela présente des avantages importants pour les utilisateurs, comme la réduction des coûts d'investissement, et pour les fournisseurs de logiciels, comme la simplification du déploiement des nouvelles versions des logiciels.
- Les systèmes multitenants sont des systèmes SaaS dans lesquels tous les utilisateurs partagent la même base de données, qui peut être adaptée en cours d'exécution à leurs besoins individuels. Les systèmes multi-instances sont des applications SaaS dans lesquelles chaque utilisateur possède sa propre base de données.
- Les principaux aspects architecturaux des logiciels en cloud sont la plateforme en cloud à utiliser, l'utilisation d'une base de données multitenant ou multi-instance, les exigences en matière d'évolutivité et de résilience, et l'utilisation d'objets ou de services en tant que composants de base du système.