Engineering Simplicity

Livre blanc

ARCHITECTURE JUNIPER APSTRA

Résoudre des problèmes difficiles de centre de données grâce à une automatisation sur mesure pour l'ensemble du cycle de vie du réseau

TABLE DES MATIÈRES

SYNTHÈSE

Construire, déployer, opérer, gérer et dépanner un réseau de centre de données peut s'avérer difficile, onéreux et gourmand en ressources. Selon le rapport Gartner de 2019, simplifier l'infrastructure informatique est aujourd'hui la principale priorité stratégique des entreprises. Juniper® Apstra a été conçu sur mesure pour automatiser l'ensemble du cycle de vie du réseau de centre de données.

Ce livre blanc décrit certains des plus grands défis actuels auxquels doivent faire face les équipes chargées du réseau de centre de données et explique comment l'architecture d'Apstra résout chacun d'entre eux.

Introduction

Les équipes chargées de l'architecture et des opérations réseau du centre de données doivent faire face à quatre difficultés majeures :

- **Composition**. Assembler un système cohérent qui fonctionne sans problème avec des composants d'infrastructure provenant de différents fournisseurs et ayant des capacités propres peut être incroyablement difficile. Juniper® Apstra crée un ensemble cohérent (un blueprint) composé de toutes les informations nécessaires pour déployer le système en fonction de l'intention exprimée. Le blueprint est transmis à l'infrastructure physique et permet à un opérateur de fournir des services fiables et faciles à utiliser.
- **Changements fiables**. Les changements sont une constante dans le centre de données. Un opérateur peut vouloir ajouter de nouveaux services ou développer la capacité, ou l'infrastructure peut changer en fonction de conditions opérationnelles de défaillance. Dans les deux cas, l'opérateur doit trouver de nouvelles manières de gérer ces changements. Selon plusieurs études réalisées ces dernières années[1], 65 % à 70 % des pannes réseau sont provoquées par des erreurs de configuration humaines lors de l'exécution d'un changement planifié.
- **Extensibilité**. Pour encourager l'innovation et faciliter les changements technologiques rapides inévitables, les centres de données doivent pouvoir ajouter de nouvelles capacités et fonctionnalités facilement et sans problème. À mesure que la technologie évolue, les fournisseurs innovent et proposent de nouvelles fonctionnalités dont les opérateurs peuvent tirer profit pour rester compétitifs. Cette évolution se reflète sur la conception, avec un besoin pour de nouveaux contrats comportementaux (ou conceptions de référence) pour régir les services que vous souhaitez déployer. En d'autres termes, votre composition doit être extensible pour permettre d'autres innovations ou répondre à de nouvelles exigences.
- **Évolutivité**. Pour répondre à tout cela à grande échelle, les réseaux de centre de données doivent pouvoir s'étendre, servir de moteur aux innovations et s'adapter à la complexité croissante.

Le premier objectif de l'architecture d'Apstra est de traiter directement ces problèmes.

La problématique principale : composition

La principale difficulté que rencontrent les opérateurs exploitant actuellement une infrastructure de calcul/réseau/stockage, c'est la composition. La composition est la capacité à créer un ensemble cohérent plus grand que la somme de ses composants et de maintenir cette cohérence dans le temps malgré les changements inévitables, tout en fournissant des services réseau aux consommateurs. Apstra a été essentiellement conçu pour répondre au défi de composition.

Les centres de données modernes sont des ordinateurs à évolutivité horizontale. Ils ont besoin d'un système d'exploitation qui fournit des fonctionnalités analogues à celles qu'un système d'exploitation hôte fournit sur une machine unique aujourd'hui : gestion des ressources et isolation des processus.

La virtualisation des serveurs assure déjà ces fonctions dans les configurations à un seul serveur. Cependant, dans les centres de données à évolution horizontale, l'opérateur doit d'abord les composer avant de partitionner les ressources.

[1] Exemples : https://www.computerweekly.com/news/2240179651/Human-error-most-likely-cause-of-datacentre-downtime-finds-study ; https://www.networkworld.com/article/3142838/top-reasons-for-network-downtime.html; https://www.ponemon.org/library/national-survey-on-data-center-outages.

Pourquoi la composition est-elle si difficile ?

Deux raisons principales expliquent cette difficulté. Premièrement, les éléments utilisés pour composer un ensemble cohérent peuvent provenir de différents fournisseurs, offrir différentes capacités ou n'être disponibles que via différentes API.

Deuxièmement, vous pouvez avoir besoin de composer votre infrastructure pour fournir divers services, chacun avec plusieurs aspects fonctionnels tels que l'accessibilité, la sécurité, la qualité d'expérience et la disponibilité.

Lorsque vous instanciez plusieurs instances de service, des interactions entre ces composants peuvent entraîner des pannes de service, à moins que vous sachiez précisément comment mapper ces services en mécanismes d'application. À mesure que les capacités de votre infrastructure évoluent, vous devez pouvoir introduire et exploiter des innovations sans déranger les systèmes existants.

Apstra relève ces défis avec les conceptions de référence

Le concept clé de l'architecture Apstra pour relever la difficulté de composition est la conception de référence. Une conception de référence est un contrat comportemental qui définit comment l'intention est mappée à des mécanismes d'application et les attentes qui doivent être satisfaites pour considérer que l'intention est réalisée.

Les conceptions de référence régissent :

- Les rôles et responsabilités des composants physiques et logiques
- La manière dont les services sont mappés aux mécanismes d'application
- Les attentes qui doivent être satisfaites (c'est-à-dire les situations à surveiller)

Dans une conception de référence, les rôles et les responsabilités des éléments physiques et logiques sont strictement définis, ce qui établit la portée de la modélisation et précise la spécification d'un modèle minimal mais complet. Une conception de référence régit également la manière dont l'intention est mappée aux mécanismes d'application. La compréhension de ce mappage permet d'automatiser le dépannage et d'offrir des analyses puissantes basées sur la connaissance de la composition du système. Ainsi, plus besoin de rétro-ingénierie sur ce qui se passe dans votre réseau.

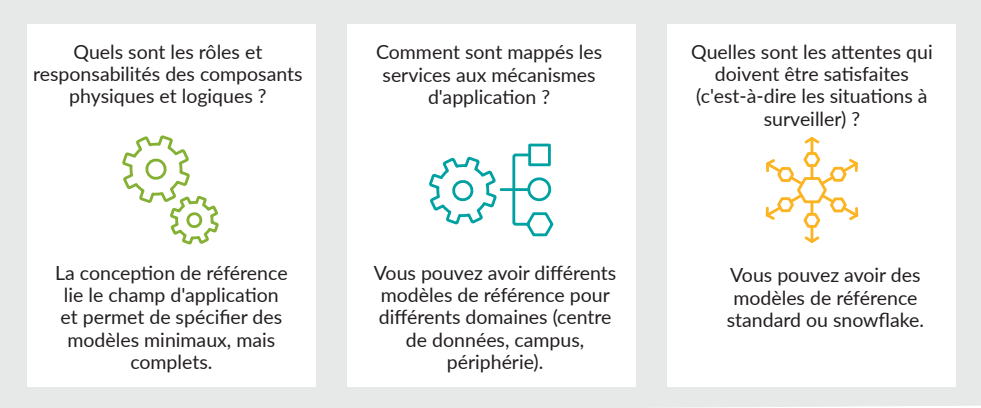

Figure 1 : La conception de référence est un contrat comportemental conducteur.

Dans différentes conceptions de référence, la même intention peut être appliquée via différents mécanismes plus ou moins récents et innovants. En comprenant ce mappage, les opérateurs peuvent directement identifier la cause racine qui impacte le service.

La conception de référence spécifie également les attentes à satisfaire. Par exemple, une conception de référence peut stipuler que chaque branche doit avoir une session BGP pour chaque équipement central. De cette manière, les sessions BGP manquantes ou inattendues sont facilement identifiables.

Définir une conception de référence est une activité à valeur ajoutée qui doit être effectuée par un expert des réseaux. Avec Apstra, l'expertise est une partie explicite du système plutôt qu'un élément qui n'existe que dans la tête de l'expert. Cela permet à l'expertise d'être modélisée et insérée dans le système, contrairement à un codage en dur.

Savoir, c'est pouvoir : faire face aux changements en toute fiabilité

> *« Les espèces qui survivent ne sont pas les espèces les plus fortes, ni les plus intelligentes, mais celles qui s'adaptent le mieux aux changements. »*
>
> - Charles Darwin, 1809

Les changements sont une constante dans les centres de données, et y faire face de manière fiable est toujours une priorité. Ils ont deux origines :

1. **L'intention de l'opérateur**. Un opérateur peut vouloir ajouter un nouveau service tel qu'un réseau virtuel, ajouter/supprimer une ressource de l'infrastructure ou modifier certaines stratégies. L'opérateur contrôle ces changements, car il les initie. Apstra utilise l'orchestration à états pour appliquer les intentions des opérateurs avec fiabilité.
2. **Des défaillances dans une infrastructure gérée.** Les changements peuvent également provenir de l'infrastructure gérée sous forme de défaillances (par exemple, des chutes de paquet excessives ou des déséquilibres de trafic). Un opérateur ne contrôle pas ces changements, et les conditions opérationnelles sont souvent extrêmement nombreuses. Apstra fournit des analyses basées sur l'intention pour aider les opérateurs à gérer les changements à l'état opérationnel.

Traiter les deux types de changement en toute fiabilité dépend de votre connaissance de l'état de votre infrastructure. Deux conditions doivent être remplies :

1. Les connaissances doivent être interprétées en contexte, ce qui signifie que la relation entre les conditions et les attentes est comprise, comme décrit dans la section Modèle de contexte.
2. Les connaissances doivent être opportunes, c'est-à-dire qu'elles doivent refléter les conditions actuelles, comme décrit dans la section Surveillance et notifications en temps réel.

Orchestration à états

Une orchestration à états rend fiabilise changements provenant de l'opérateur grâce à l'intention. Vous pouvez voir un flux d'orchestration à états sur la Figure 2.

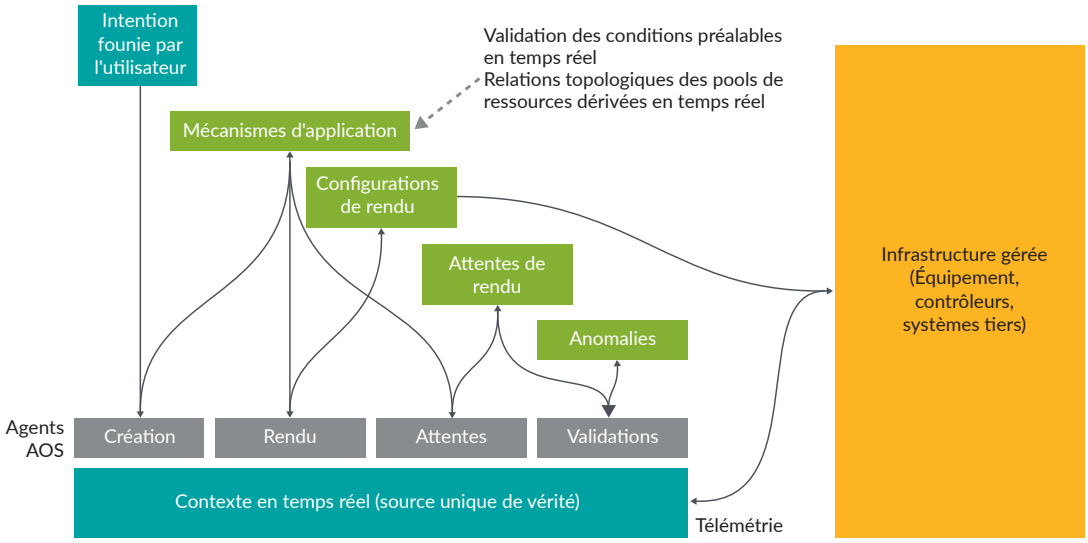

Figure 2 : Flux d'orchestration à états

Avec l'orchestration à états, un utilisateur n'a qu'à indiquer son intention pour apporter des modifications au système existant en fonctionnement. Le processus étant indépendant de toute implémentation, il est moins enclin aux erreurs et facilite l'expression des intentions. L'orchestration à états doit nécessairement suivre certaines étapes :

Étape 1 : Validation de la condition préalable en temps réel. Cette nouvelle requête va-t-elle enfreindre certaines stratégies ? Par exemple, êtes-vous autorisé à créer ce réseau virtuel, ou cela va-t-il créer des failles de sécurité ? Ou, pouvez-vous placer un équipement en mode maintenance en sachant que d'autres équipements sont déjà hors ligne et que mettre cet équipement hors ligne va rendre votre système vulnérable ? Grâce à l'orchestration à états, toutes ces questions sont posées et validées en temps réel par rapport à un graphique de contexte.

Étape 2 : Déduction des paramètres d'application en temps réel. Pour réaliser l'intention, le système doit paramétrer précisément les mécanismes d'application adéquats sur certains éléments gérés. Grâce à une conception de référence qui définit la manière d'implémenter l'intention, et à sa compréhension du contexte actuel, le système Apstra déduit les paramètres d'application en temps réel. Ainsi, l'opérateur ne peut pas saisir accidentellement une commande, une interface, une adresse IP ou un VLAN erronés.

Étape 3 : Déploiement de configurations multiples en temps réel. Apstra déploie automatiquement des configurations multiples en temps réel sur les divers éléments qui utilisent des API spécifiques au fournisseur ou à la technologie.

Étape 4 : Génération d'attentes en temps réel. Avec une conception de référence agissant comme un contrat comportemental, Apstra peut générer des attentes en temps réel qui décrivent les conditions devant être satisfaites pour considérer que le résultat a répondu à l'intention.

Étape 5 : Validation des attentes en temps réel. Apstra déclenche ensuite une collection d'analyses opérationnelles basées sur la télémétrie et le contexte pour valider les attentes en temps réel.

Étape 6 : Résultat de service validé. À la fin du processus, l'utilisateur obtient un résultat de service validé dans des termes sans ambiguïté. Chaque modification de l'intention ou du statut opérationnel attendu est reflétée dans le modèle de contexte en temps réel, et tous les composants devant être avertis du changement sont également notifiés en temps réel. Apstra extrait des connaissances pertinentes des données opérationnelles brutes à l'aide des analyses basées sur l'intention.

L'orchestration sans état n'est pas capable d'exécuter beaucoup de ces étapes (comme illustré sur la Figure 3). Ce type d'orchestration se concentre traditionnellement sur le résultat du déploiement des configurations : transmettre la configuration sur plusieurs systèmes et confirmer que les configurations ont été acceptées par les éléments gérés. C'est là que l'orchestration sans état s'arrête en général. Les attentes de service n'existent pas, pas plus que la validation de ces attentes. À la place de la validation automatisée des résultats de service, des systèmes distincts présentent à l'utilisateur des « interfaces uniques » affichant l'état opérationnel brut. C'est à l'utilisateur d'effectuer une recherche visuelle pour voir si le résultat a été atteint. Cette interface unique inclut généralement trop d'informations et manque de contexte spécifique à la mise en œuvre, ce qui rend la recherche visuelle extrêmement difficile et subjective.

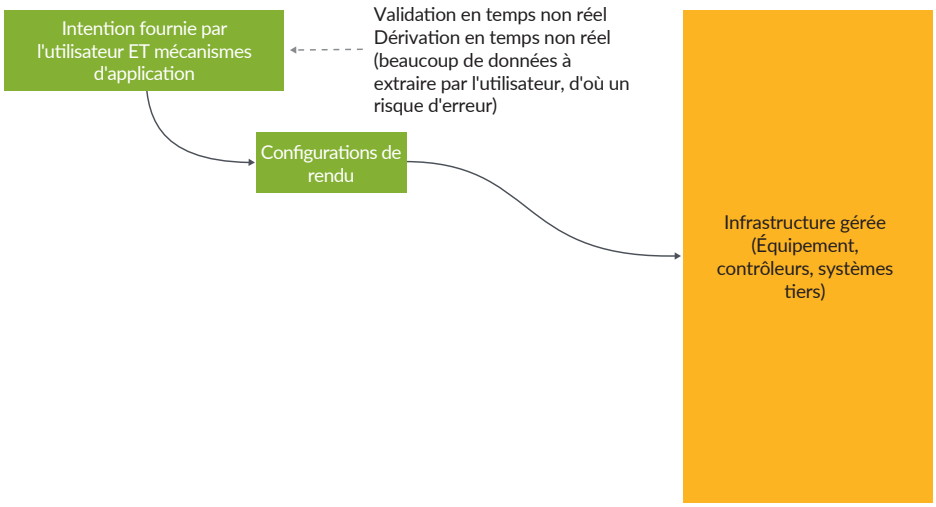

Figure 3 : Orchestration sans état classique

Analyses basées sur l'intention

L'analyse basée sur l'intention (IBA) aide les opérateurs à traiter les changements de statut opérationnel dans leur infrastructure en extrayant des connaissances des données de télémétrie brutes. Comme mentionné précédemment, vous devez avoir un contexte pouvant être interrogé en temps réel avant de pouvoir utiliser l'IBA.

L'extraction des connaissances est composée de deux étapes :

1. **Détection des conditions d'intérêt** (situations à surveiller). Les sondes IBA détectent les conditions.
2. **Automatisation de la classification des conditions d'intérêt et déduction de leurs relations**. Les conditions varient dans leur contenu sémantique. En d'autres termes, certaines sont plus importantes que d'autres. Il est essentiel de comprendre les relations entre les conditions afin de pouvoir distinguer les conditions importantes sur lesquelles agir (causes racines) de celles qui disparaîtront lorsque les causes racines auront été traitées. La classification des conditions et la déduction des liens de causalité sont effectuées par le composant d'identification de la cause racine de l'IBA.

Conditions d'intérêt de modélisation

L'IBA modélise les conditions d'intérêt en fonction de quatre catégories : anomalie, symptôme, impact et cause racine.

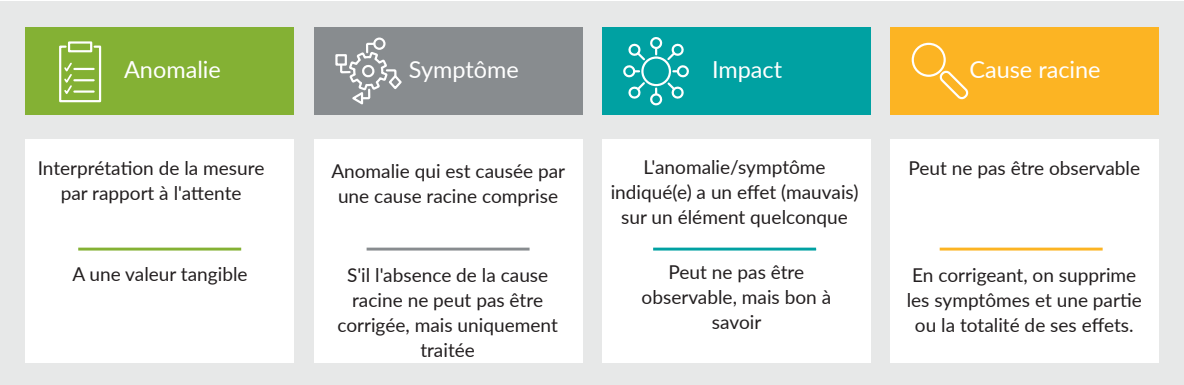

Figure 4 : Catégorisation des conditions

Anomalies

Les anomalies sont en quelque sorte une interprétation ou une agrégation de mesures par rapport à des attentes. Par conséquent, elles ont une valeur plus concrète que des mesures simples.

Symptômes

Les symptômes sont des anomalies provoquées par une cause racine bien comprise. En règle générale, ils sont facilement observables, mais ne peuvent pas être corrigés, seulement traités, de la même manière qu'on donnerait de l'aspirine à un patient pour traiter sa fièvre, mais pas la maladie même. Si vous ne traitez pas la maladie, la fièvre reviendra. Ils ne peuvent disparaître que si la cause racine est corrigée. Ils ne sont donc pas exploitables ; ils sont simplement utiles pour le diagnostic de la cause racine.

Impacts

Les impacts indiquent que quelque chose s'est produit suite à une anomalie. Les impacts ne sont pas toujours observables, mais il est utile de les connaître. Par exemple, il est utile de savoir qu'un client important va être impacté par la défaillance d'un équipement ou une perte de paquets excessive avant qu'il ne vous appelle pour se plaindre.

Lorsque les impacts ne sont pas observables, ils peuvent être calculés en fonction des connaissances de l'intention et des mécanismes d'application utilisés pour mettre en œuvre l'intention. Comprendre les impacts aide les opérateurs à prioriser quelles causes racines et quelles anomalies ont la plus grande influence sur l'opération.

Causes racines

Les causes racines sont les plus importantes des conditions. Elles ne sont pas toujours observables, ce qui complique le diagnostic, mais elles sont responsables de nombreux symptômes et impacts. Elles sont exploitables, car corriger la cause principale supprime les symptômes et impacts associés.

Les sondes IBA déterminent ce que vous savez

Une célèbre citation qui s'applique aux réseaux de centre de données : « Le danger, ce n'est pas ce qu'on ignore, c'est ce que l'on tient pour certain et qui ne l'est pas. » L'intention caractérise et formalise la normalité, ce que nous savons. L'IBA garantit que ce que nous savons est effectivement ce qui devrait être.

> *« Le danger, ce n'est pas ce qu'on ignore, c'est ce que l'on tient pour certain et qui ne l'est pas. »* - Mark Twain

Les sondes IBA détectent les conditions d'intérêt. Elles sont basées sur un contrat comportemental, contenu dans une conception de référence. Ces sondes récupèrent des données, appliquent un traitement, puis comparent le résultat aux attentes. En bref, une sonde IBA est un pipeline configurable de traitement des données qui permet aux utilisateurs de définir les conditions d'intérêt (c'est-à-dire les situations à surveiller).

Sources de données et requêtes

Les sondes démarrent généralement avec des sources de données classiques, qui récupèrent des données de télémétrie brutes. Pour configurer de cette étape, une fonctionnalité très puissante de requête permet d'exprimer précisément les données à collecter.

Par exemple, supposons qu'un opérateur souhaite analyser un déséquilibre dans le routage multi-chemin à coût égal (ECMP) qui s'applique uniquement aux interfaces de la structure, et que des rapports ont signalé qu'une version de système d'exploitation spécifique a introduit un bogue dans l'algorithme de hachage de l'ECMP. Une requête peut exprimer le besoin de collecter les compteurs d'interfaces sur chaque commutateur haut de baie, mais uniquement pour les interfaces de structure et pour les commutateurs qui exécutent une version particulière x.y.z du système d'exploitation. Maintenant que la situation à surveiller est définie, l'opérateur n'a pas besoin de se soucier des changements. Si une nouvelle liaison de structure est ajoutée à un commutateur qui correspond au critère, elle sera automatiquement incluse dans l'analyse. Si un nouveau commutateur est ajouté, il sera automatiquement inclus. Si quelqu'un met à niveau un autre commutateur à la version x.y.z, il sera automatiquement inclus. Les sondes IBA ne nécessitent pas de maintenance.

Processeurs d'étape

Pour les opérateurs, les valeurs ponctuelles des données de télémétrie brutes sont souvent moins intéressantes que les valeurs agrégées ou les tendances. L'IBA contient des processeurs d'étape qui agrègent les informations, par exemple en calculant la moyenne, les min./max., la déviation standard, etc. Un opérateur peut ensuite comparer ces agrégats avec les attentes : s'ils ne sont pas dans la plage spécifiée, une anomalie est signalée.

L'opérateur peut ensuite vouloir déterminer si cette anomalie dure plus longtemps qu'un seuil spécifique, et signaler l'anomalie seulement dans ce cas pour écarter les conditions transitoires ou temporaires. Pour obtenir ce résultat, l'opérateur peut simplement configurer une étape secondaire avec un processeur de durée d'état.

Des données numériques et plus encore

Les sondes ne fonctionnent pas seulement avec données numériques. Par exemple, elles peuvent être utilisées pour valider l'exactitude des plans de données et de contrôle. Un opérateur peut, par exemple, configurer un processeur de source de données avec une requête pour construire une table de transfert ou de routage attendue, en fonction de l'intention.

Dans un environnement EVPN (Ethernet VPN), l'intention contiendra des informations telles que les réseaux virtuels existants, l'emplacement de leurs points de terminaison et les informations relatives aux mécanismes d'application. Cette table attendue peut ensuite être comparée à celle provenant de la télémétrie, pour signaler les anomalies en cas d'incohérence.

Une sonde peut également être configurée pour suivre les changements de taille soudains dans les tables de transfert et de routage lorsque les tendances ne correspondent pas aux attentes. Le seuil de « ce qui est attendu » peut également être calculé dynamiquement à partir de l'intention, car il peut être une fonction du nombre de réseaux virtuels, du nombre de points de terminaison, du nombre de points de terminaison de tunnels virtuels (VTEP), du nombre de VTEP ayant des problèmes, etc. Les possibilités sont infinies.

Les sondes peuvent être créées, activées et désactivées de manière déclarative à l'aide d'un simple appel REST ou en utilisant une interface utilisateur graphique (GUI). Activer les sondes permet également de déclencher la télémétrie. En d'autres termes, des données de télémétrie spécifiques sont collectées uniquement si elles sont demandées par une sonde. Les sondes sont donc un mécanisme pour configurer la collecte de télémétrie. Les requêtes dans les processeurs de source de données rendent la configuration aussi précise et définie que nécessaire. Ainsi, plus besoin d'accumuler des tonnes données sans raison précise. Les coûts de stockage et de traitement diminuent donc drastiquement.

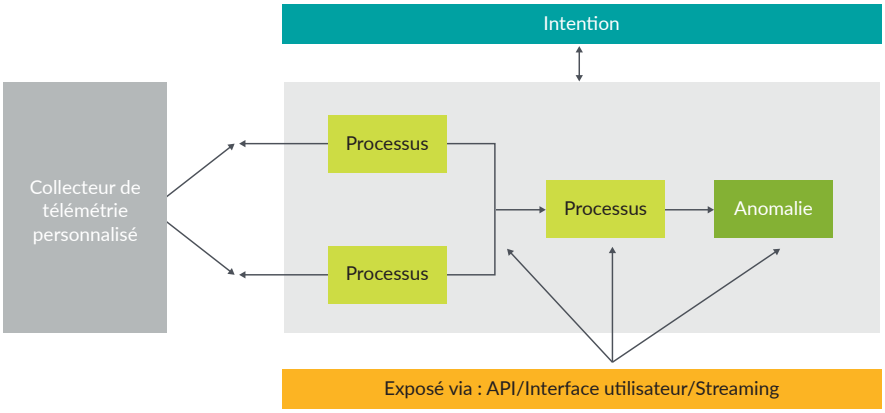

Figure 5 : Collecte de télémétrie

Lorsqu'une sonde est configurée, les sorties de toutes les étapes sont disponibles via des API, une interface utilisateur ou encore des points de terminaison de streaming. Apstra comporte un ensemble de sondes intégrées, ainsi que d'un référentiel de sondes open source sur GitHub. L'utilisateur peut également créer ses propres sondes en partant de zéro.

Identification de la cause racine

L'identification de la cause racine (RCI) est un mécanisme permettant d'automatiser la classification et la déduction des relations de causalité entre les conditions identifiées dans l'infrastructure. L'avantage principal de la RCI est d'identifier les conditions de la cause racine qui nécessitent une action de la part de l'opérateur parmi toutes les conditions non exploitables qui ne sont que des conséquences de la cause racine. Par exemple, supposons qu'un opérateur a bien instrumenté l'infrastructure, et qu'il consulte un journal de conditions d'anomalies sur une console centralisée (voir Figure 6). Les points rouges indiquent les différents types de conditions répartis sur l'infrastructure, survenant sur plusieurs éléments différents.

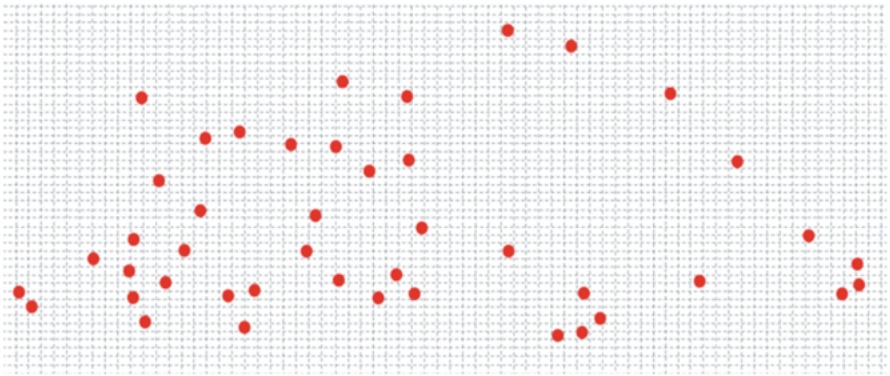

Figure 6 : Console centralisée

Le problème de cette image est qu'elle n'est pas exploitable. C'est essentiellement du bruit. En l'absence de classification, chaque condition attire l'attention de l'opérateur, mais lesquelles traiter ? C'est à ce moment que la RCI entre en jeu. Grâce à la connaissance du contexte, qui indique la nature du service, son implémentation et son mappage dans les mécanismes d'application, ainsi que les relations entre les éléments de l'infrastructure gérée, la RCI identifie et classifie la ou les causes racines et les symptômes et impacts associés. Comme illustré sur la Figure 7, les résultats de l'analyse utilisant la RCI sont les suivants :

- La cause racine était une fuite de mémoire sur un commutateur nommé « spine_1 » (cause racine : point le plus large).
- Cette fuite de mémoire a poussé le tueur de processus à terminer des processus manquant de mémoire, dont le processus de routage (symptômes : petits points foncés).
- Par la suite, cela a entraîné l'échec des sessions BGP sur cet équipement et les équipements appairés, ainsi que des entrées de table de routage manquantes (symptômes : petits points foncés).
- Résultat : les points de terminaison appartenant aux clients X et Y rencontrent des problèmes de connectivité (impacts : petits points clairs).

Figure 7 : Console d'identification de la cause racine

La RCI automatise le processus mental complexe associé aux interfaces centralisées, qui surchargent les opérateurs avec une multitude d'informations à trier et corréler visuellement. À la place, la RCI présente à l'opérateur une interface simple qui identifie les actions nécessaires.

Modèle de contexte

Avoir toutes les données ne signifie pas avoir toutes les réponses. En outre, collecter de grandes quantités de données sans automatiser l'extraction des connaissances ne fait qu'augmenter les coûts d'exploitation, car vos experts passeront un temps précieux à essayer de déchiffrer ces données. Les données en elles-mêmes n'ont qu'une valeur limitée, à moins qu'elles ne vous apportent des connaissances ou des réponses aux bonnes questions.

Les données de télémétrie brutes sont généralement stockées dans des magasins de paires clé/valeur, adaptés à l'évolution horizontale et au sharding. Cependant, les seules requêtes acceptées sont les recherches de clés simples. Pour extraire des connaissances, il faut prendre en charge des requêtes plus complexes, avec un contexte pour les construire.

D'un autre côté, les magasins de données SQL prennent en charge des requêtes complexes. Le problème est que les tables SQL sont conçues autour de requêtes anticipées, qui sont créées pendant la conception. Si vous souhaitez poser d'autres questions pendant l'exécution, vous pouvez dénormaliser la base de données. Dans le cas contraire, les requêtes pourraient avoir de nombreuses liaisons, ce qui paralyserait les performances. Mais pour extraire des connaissances opérationnelles, les requêtes doivent porter sur l'exécution. L'implémentation graphique des données contextuelles entre en scène : elle prend en charge des requêtes complexes arbitraires avec un grand nombre de « liaisons » pendant l'exécution. Le modèle graphique peut également être étendu facilement en ajoutant simplement plus d'instances de blocs modulaires, de nœuds et de relations.

Examinons les éléments qui composent ce contexte pour expliquer la puissance des requêtes.

- **Artefacts de conception** : tout d'abord, le contexte intention-graphique contient des artefacts de conception. Par exemple, si vous concevez un réseau de centre de données, il peut contenir le facteur de surabonnement souhaité, le nombre de serveurs souhaités et la configuration haute disponibilité (HA) souhaitée (liaison unique, liaison double).
- **Attribution des ressources** : le contexte contient les décisions associées à l'attribution des ressources. Il détermine si votre infrastructure utilise plutôt du « bétail » ou des « animaux de compagnie ».
- **Stratégies d'isolation** : il spécifie également les stratégies d'isolation. Peut-on réutiliser certaines ressources (adresses IP, ASN, VLAN) ?
- **Stratégies de segmentation** : il contient les stratégies de segmentation qui spécifient quels points de terminaison et quelles charges de travail peuvent communiquer et dans quelles conditions.
- **Informations d'application** : il contient les informations d'application qui déterminent où (quels équipements physiques ou virtuels ?) et comment (quels mécanismes ?) sont implémentées les stratégies de segmentation et d'accessibilité.
- **Stratégies d'accessibilité externe** : dans le cas du centre de données, il peut également contenir les stratégies d'accessibilité externes, spécifiant quels points de terminaison internes peuvent être découverts en externe, et vice versa. Il spécifie également quels locataires possèdent quels segments, les niveaux de service promis à chacun d'entre eux et les objectifs de niveau de service.
- **Perspective métier** : toutes les informations métier sont incluses, par exemple les accords de niveau de service et les coûts encourus s'ils ne sont pas remplis.

Vue 8-D

Même si tous ces artefacts sont représentés dans un graphique unique, cette source unique de vérité est un espace multidimensionnel unique qui comporte, au minimum, les huit dimensions suivantes (choisies pour cet exemple ; davantage sont possibles) :

- Conception
- Ressources
- Isolation
- Segmentation
- Application
- Accessibilité externe
- Niveaux de service
- Perspective métier

Résultat : en cas de problème, une unique requête peut naviguer cette vue 8-D pour répondre à des questions complexes comme : « Étant donné la récente condition de défaillance sur un élément, mes objectifs de conception sont-ils toujours valides ? » « Est-ce que les bonnes ressources sont utilisées sur les bons points d'application, avec des stratégies d'isolation adéquates ? » « Est-ce que cela impacte la stratégie de segmentation d'une quelconque manière ? » « Est-ce que l'objectif de niveau de service sera respecté ? » « Quel prix dois-je payer pour cette panne ? »

Cette vue 8-D n'est pas une fonctionnalité « pratique » ; c'est un prérequis pour la fiabilité, qui est intégré au système basé sur l'intention Apstra. Sans Apstra, vous devez reconstruire cette vue 8-D sous forme de couche très complexe, en prenant en compte plusieurs sources fiables et des connaissances qui sont uniquement connues par certains de vos experts. Construire cette couche dans un environnement où les sources uniques de vérité sont difficiles à intégrer, avec des natures et des comportements multiples, représente un effort fastidieux et peu productif dans le meilleur des cas, et un cauchemar ingérable dans le pire des cas.

Surveillance et notifications en temps réel

Apstra extrait des connaissances sur vos intentions et sur l'état opérationnel de votre infrastructure correspondant. Ces connaissances sont uniquement utiles si elles sont opportunes, c'est-à-dire si elles reflètent les conditions actuelles. En essence, Apstra permet de configurer les requêtes en direct, ce qui permet aux clients de s'abonner aux conditions d'intérêt et d'être notifiés en temps réel lorsque les conditions sont satisfaites. Les conditions, dans ce contexte, sont associées à l'intention et à l'état opérationnel. Il s'agit d'un prérequis absolu pour traiter les changements en toute fiabilité.

Comme vous le verrez dans la section Présentation de l'architecture, le mécanisme « pub/sub » (publish/subscribe) pour l'utilisateur trouve un équivalent de bas niveau. Ce mécanisme est mis en œuvre par le magasin de données distribué agissant comme un canal de communication logique centré sur les données pour les processus du système Apstra qui mettent en œuvre la logique d'application.

Fonctionnement autonome

Enfin, il n'y a pas besoin d'automatiser la réaction à certains événements, que ce soit pour résoudre les problèmes, les enregistrer pour analyse ultérieure, ou collecter des données télémétriques approfondies pour identifier la cause racine.

Une réaction automatisée :

- réduit les risques, car les paramètres de résolution sont automatiquement dérivés d'une source unique de vérité actualisée et ne peuvent pas être mal configurés ou utiliser des données obsolètes ;
- améliore l'expérience du client, puisque la résolution survient de manière opportune ;
- réduit les coûts opérationnels, car elle supprime le besoin d'un dépannage manuel et l'exécution manuelle du guide stratégique de résolution.

Les difficultés d'un réseau autonome sont davantage liées à une résistance organisationnelle ou individuelle qu'à des limites technologiques, car les fonctionnalités adéquates existent déjà.

Au final, une réaction automatisée permet au réseau de fonctionner et de se réparer de manière autonome. Les difficultés d'un réseau autonome sont davantage liées à une résistance organisationnelle ou individuelle qu'à des limites technologiques, car les fonctionnalités adéquates existent déjà.

Extensibilité : pérenniser le réseau de centre de données

L'extensibilité a trois dimensions :

1. **Extensibilité de la conception de référence**. Elle régit la manière dont les pièces de l'infrastructure collaborent pour réaliser l'intention. Elle inclut des modules complémentaires permettant de modifier le modèle graphique ainsi que le mappage intention-mécanismes d'application au sein de l'infrastructure.
2. **Extensibilité analytique**. Elle correspond à l'ajout de nouvelles conditions ou de situations à surveiller, et permet à l'utilisateur de définir de nouvelles validations ainsi que la manière de les classifier et de les associer.
3. **API de service flexibles**. Elles permettent d'étendre les définitions de service de premier niveau.

Conception de référence

Comme mentionné plus tôt, une conception de référence est un contrat comportemental qui définit comment l'intention est mappée à des mécanismes d'application et les attentes qui doivent être satisfaites pour considérer que l'intention est réalisée. Tout aspect de ce contrat peut être modifié ou étendu. Il est possible de définir de nouveaux types de nœuds et de relations, et de modifier les modèles de configuration et les mécanismes d'attribution des ressources.

Extensibilité analytique

De nouvelles fonctionnalités analytiques peuvent être introduites par le biais de deux mécanismes :

1. De **nouvelles sondes IBA** peuvent être définies pour détecter de nouvelles conditions d'intérêt. Elles peuvent inclure de nouveaux collecteurs de télémétrie, ainsi que des pipelines de traitement des données spécifiques pour les différentes conditions. Les analyses peuvent être publiées, puis importées à partir d'un référentiel public.
2. De **nouveaux modèles de RCI** peuvent être définis et chargés dans un système. Dans l'absolu, un modèle de RCI mappe un nouveau type de cause racine à un ensemble de symptômes qui en résultent. Une fois ce modèle défini et chargé, le système Apstra automatise l'identification de la cause racine en fonction des symptômes observés.

API de service

Apstra expose les API des services sous forme de stratégies de groupes, ce qui permet de prendre en charge une large gamme de services et de stratégies indépendamment de l'implémentation, pour plus de flexibilité.

Avec les stratégies de groupe, l'intention est traduite en graphique : les points de terminaison sont placés dans des groupes (relations de membre) que l'intention associe à des comportements communs. Les stratégies sont instanciées et associées à des groupes ou des points de terminaison individuels pour définir ce comportement.

Elles peuvent être liées à un groupe de manière directionnelle (relation de/à) ou non directionnelle (s'applique à). Les stratégies sont une collection de règles. Les règles peuvent déclencher une autre règle lorsque leur ordre est important. Les groupes peuvent être constitués d'autres groupes (hiérarchie). Les groupes peuvent également avoir une relation avec d'autres groupes pour exprimer des contraintes (telles que « ces points de terminaison/groupes se trouvent derrière ces groupes de ports »). Les points de terminaison, les groupes, les stratégies et les règles peuvent être considérés comme des blocs modulaires qui expriment l'intention de connectivité.

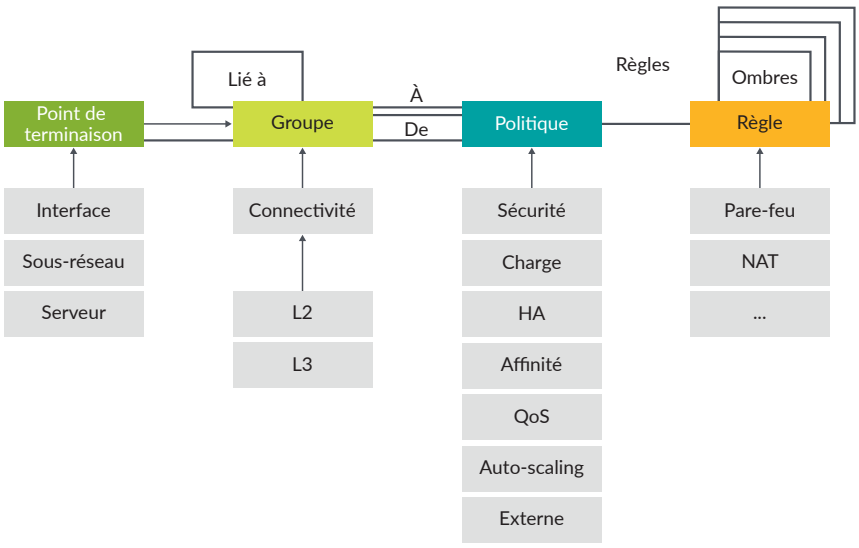

Figure 8 : Stratégies de groupe

Les points de terminaison sont des éléments de votre infrastructure qui sont soumis à des stratégies. Par conséquent, ils ne nécessitent pas de construction précise. Ils peuvent représenter une interface (physique ou virtuelle), un serveur, une machine virtuelle (VM), un conteneur ou un point de terminaison d'application. Les flèches bleues dans la Figure 8 indiquent un héritage « logique ». Les points de terminaison contiennent des paramètres qui les définissent plus précisément (par ex. nom d'interface, numéro de port, numéro de série et nom d'hôte, UUID de VM, adresse IP de conteneur, protocole d'application, numéro de port UDP/TCP). Les points de terminaison peuvent également représenter des éléments non gérés par Apstra (points de terminaison externes) et, par conséquent, sont utilisés pour exprimer des contraintes provenant des systèmes externes avec lesquels Apstra doit interagir (par ex. adresse IP/ASN de routeur externe).

Les points de terminaison sont placés dans des groupes. Tout point de terminaison donné peut être membre de plusieurs groupes, chacun exprimant différents aspects d'un comportement prévu. Par exemple, un groupe peut indiquer :

- un ensemble de serveurs dans la région « Est » ;
- un groupe de serveurs utilisés pour l'équilibrage de charge ;
- un groupe de serveurs formant un groupe de redondance.

Les groupes peuvent également contenir des implications sémantiques. Par exemple, les points de terminaison membres du groupe de domaine L2 sont membres d'un domaine de diffusion L2 (sous-réseau). De même, les points de terminaison membres du domaine L3 ont une accessibilité L3.

Les groupes peuvent également être constitués d'autres groupes. Les points de terminaison membres du groupe peuvent être instanciés explicitement ou implicitement, via une spécification d'appartenance dynamique à un groupe.

Par exemple, un domaine L3 peut avoir un CIDR (Classless Interdomain Routing) ou un sous-réseau comme propriété ; par conséquent, tous les points de terminaison avec des adresses IP dans cette plage/ce sous-réseau sont implicitement membres du groupe. Dans ce cas, la conception de référence L3-vers-serveur spécifie un nombre de sous-réseaux externes/internes. Même si les points de terminaison (conteneurs) ne sont pas explicitement spécifiés et gérés dans Apstra, ils appartiennent implicitement aux domaines L3 spécifiés. Cependant, afin de modeler une segmentation précise, nous allons également explicitement spécifier des points de terminaison de conteneur et les serveurs sur lesquels ils sont hébergés.

Une stratégie définit un comportement commun et peut contenir des paramètres pour définir précisément ce comportement. Les stratégies peuvent être appliquées au groupe de manière non directionnelle, ou de manière directionnelle si nécessaire, comme pour les stratégies de sécurité. Exemples de stratégies spécifiques : sécurité, équilibrage de charge, haute disponibilité, affinité (par ex. le souhait de placer des points de terminaison en colocalisation ou de les distribuer) et qualité de service (QoS).

Les stratégies peuvent contenir des règles au besoin. Les règles suivent généralement le modèle « une condition entraîne une action ». Par exemple, dans la stratégie de sécurité, « correspond » est une déclaration de condition, et l'action est « autoriser/refuser/enregistrer ».

Évolutivité : grandir sans obstacle

Apstra offre un accès et une gestion distribués, transparents sur le réseau, tandis que les processus distincts prennent en charge une exécution parallèle. L'exécution en temps réel repose sur un modèle d'exécution asynchrone basé sur les événements avec une planification de l'exécution en temps réel. L'efficacité et la prévisibilité reposent sur la compilation via C++ comme langage intermédiaire pour obtenir une efficacité de niveau machine. L'évolutivité repose sur trois dimensions.

Faire évoluer l'état

La première dimension est l'évolution de l'état. Les magasins de données Apstra évoluent horizontalement en ajoutant davantage de paires de serveurs haute disponibilité. Les magasins de données d'intention et de télémétrie sont séparés et peuvent évoluer indépendamment au besoin. Les magasins de données sont aussi hiérarchiques : un magasin de données de niveau supérieur s'abonne (synchronisation) au sous-ensemble d'état requis (comme spécifié par le concepteur) du magasin de données de niveau suivant, comme requis pour corréler l'état dans l'ensemble des magasins de données.

Faire évoluer le traitement

La deuxième dimension d'évolution est le traitement. Apstra peut lancer plusieurs copies des agents de traitement (par type d'agent), si et lorsque nécessaire, qui partageront la charge de traitement. Des agents supplémentaires peuvent être ajoutés en ajoutant davantage de serveurs pour les héberger. Apstra gère également le cycle de vie d'un agent.

L'architecture « publish-subscribe » (publication et abonnement) basée sur l'état du système permet aux agents de réagir (de fournir une logique d'application) à un sous-ensemble d'états bien défini. L'ensemble de l'intention est couverte par le biais d'agents distincts délégués à traiter différents sous-ensembles d'états. Cela signifie qu'en cas de changement dans l'intention ou l'état opérationnel, la réaction de l'agent est de « changer de manière incrémentielle » indépendamment de la taille de l'état complet.

Apstra utilise l'approche classique pour gérer l'évolutivité et la complexité associée : la décomposition. L'approche « tout le monde sait tout » n'évolue pas. Il faut distribuer les connaissances sur l'état souhaité et permettre à chaque agent de déterminer comment atteindre cet état pour éviter de prendre des décisions de manière centralisée. Grâce aux requêtes graphiques en direct, les clients (interfaces utilisateurs, par exemple) peuvent demander à Apstra ce qu'ils veulent et obtenir exactement ce dont ils ont besoin. Apstra contrôle ainsi précisément la quantité de données à récupérer du back-end.

Faire évoluer le trafic réseau

La troisième dimension est de l'évolution du trafic. La communication entre les agents et le magasin de données utilise un canal binaire optimisé, réduisant ainsi considérablement la quantité de trafic par rapport à des protocoles textuels.

Pour la tolérance aux pannes, une application Apstra est exécutée en tant que processus multiples sur des équipements matériels distincts (connectés par un réseau). Elle sépare l'état du traitement et prend en charge des états dupliqués et des reprises rapides d'état.

Présentation de l'architecture d'Apstra

Dans la première partie de ce livre blanc, nous avons étudié certaines des difficultés architecturales des réseaux de centre de données et la manière dont Apstra fonctionne pour résoudre ces problèmes. Ici, nous plongeons dans les spécificités de l'architecture d'Apstra.

Apstra est basé sur une infrastructure distribuée de gestion de l'état, qui peut être décrite comme une structure de communication centrée sur les données, avec un magasin de données gardé en mémoire, tolérant aux pannes et pouvant évoluer horizontalement. Toutes les fonctionnalités de la conception de référence choisie sont implémentées par un ensemble d'agents sans état. Les agents communiquent les uns avec les autres via un canal de communication de publication et d'abonnement (publish-subscribe) et implémentent la logique de l'application.

Une conception de référence Apstra est appliquée par un ensemble simple d'agents sans état, comme décrit ci-dessus. D'une manière générale, on compte trois classes d'agents :

1. Les **agents d'interaction (Web)** sont responsables des interactions avec les utilisateurs. Ils prennent donc l'entrée utilisateur et fournissent aux utilisateurs le contexte pertinent du magasin de données.
2. Les **agents d'application** sont responsables de l'exécution des transformations des données spécifiques au domaine de l'application en s'abonnant aux entités d'entrée et en fournissant des entités de sortie.
3. Les **agents d'équipement** résident sur (ou sont des proxys pour) un système physique ou virtuel géré, tel qu'un commutateur, un serveur, un pare-feu, un équilibreur de charge ou même un contrôleur. Ils permettent de rédiger une configuration et de collecter des données de télémétrie via des interfaces natives (spécifiques à l'équipement).

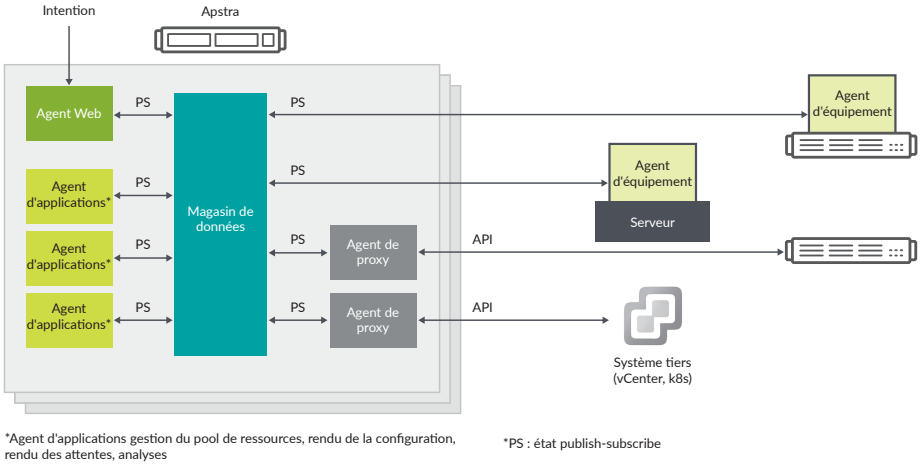

Figure 9 : Agents Apstra

Illustrons cette interaction en prenant un exemple d'application Apstra d'une conception de référence de mise en réseau de centre de données.

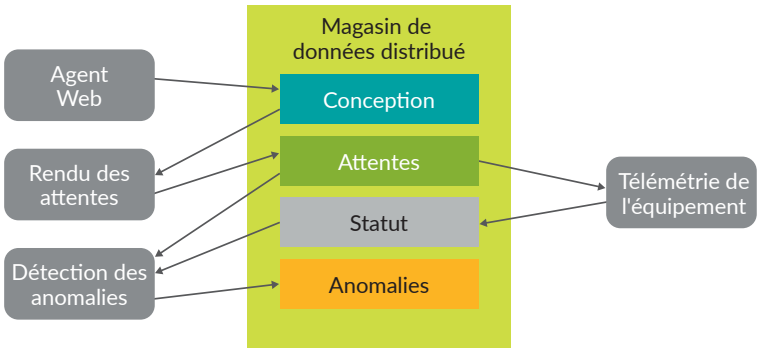

Figure 10 : Système Apstra basé sur l'intention : application d'une conception de référence de mise en réseau de centre de données

L'agent Web reçoit l'entrée utilisateur, dans ce cas, une conception pour une structure Clos L3 avec un certain nombre de cœurs, de branches et de liaisons entre eux, ainsi que les pools de ressources à utiliser pour les adresses IP de structure et les ASN (Abstract Syntax Notation). L'agent Web publie cette intention dans le magasin de données sous forme d'ensemble de nœuds graphiques, de relations et de leurs propriétés respectives.

L'agent de construction s'abonne à cette intention et :

- corrige et valide ;
- attribue les ressources des pools de ressources.

Ensuite, si les validations réussissent, l'agent de construction publie cette intention avec les attributions de ressource dans le magasin de données :

1. L'agent de rendu de la configuration s'abonne à la sortie de l'agent de construction.
2. Pour chaque nœud, l'agent de configuration récupère les données pertinentes, y compris les ressources, et les fusionne avec les modèles de configuration.
3. L'agent des attentes s'abonne également à la sortie de l'agent de construction et génère des attentes qui doivent être satisfaites afin de valider le résultat.
4. L'agent de télémétrie de l'équipement s'abonne à la sortie de l'agent des attentes et commence à collecter les données de télémétrie pertinentes.
5. Les sondes IBA traitent les données de télémétrie brutes, les comparent avec les attentes et publient les anomalies.
6. L'agent d'identification des causes racines (RCI) analyse les anomalies et les classe en symptômes, impacts et causes racines identifiées.

Les agents communiquent via des interfaces basées sur les attributs (ce qui explique le terme « centré sur les données ») en publiant des entités et en s'abonnant aux changements apportés aux entités. Le terme « centré sur les données » implique également que le cadre donne du sens aux données en définissant les entités, contrairement aux systèmes basés sur les messages, par exemple.

Le système « publish-subscribe » (publication et abonnement) n'est pas sujet aux problèmes des systèmes basés sur les messages. Dans un système basé sur les messages, tôt ou tard le nombre de messages dépasse la capacité de stockage ou de consommation du système. Ce problème est difficile à résoudre, car l'historique des messages est nécessaire pour obtenir un état cohérent.

Le système centré sur les données est résilient aux changements brusques d'état, car il dépend uniquement du dernier état. Cet état capture le contexte important et retire toutes les séquences d'événements possibles (et non pertinentes) qui y mènent. Le code écrit pour une machine à états est plus facile à lire, à maintenir et à déboguer.

Les problèmes compliqués (par ex. l'élasticité et la tolérance aux pannes) sont résolus grâce aux agents. L'architecture type se compose alors d'un nombre d'agents sans état qui peuvent être redémarrés en cas de défaillance et qui reprennent là où ils s'étaient arrêtés en relisant simplement l'état de leur abonnement dans la base de données du système (SysDB).

Avantages de l'architecture Apstra

L'architecture Apstra offre des avantages considérables, résolvant certains des problèmes de mise en réseau de centre de données les plus difficiles, comme nous l'avons expliqué dans ce livre blanc. L'architecture :

- **aide les opérateurs à gérer les changements de manière fiable**. C'est possible grâce à des intentions et contextes opérationnels pouvant être interrogés en temps réel.
- **simplifie tous les aspects du cycle de vie des services réseau**, y compris les opérations des jours 0, 1 et 2. La simplicité réduit les risques d'erreur de la part de l'opérateur.
- **réduit les risques opérationnels par le biais d'une orchestration à états** qui s'appuie sur des validations préalables et postérieures, sur le rendu automatisé des configurations et sur des validations automatisées d'attentes.

Avantages particuliers de l'identification de la cause racine et des analyses basées sur l'intention

Les composants analytiques opérationnels d'Apstra (identification de la cause racine et analyses basées sur l'intention) vous permettent de :

- **réduire le temps moyen de réparation et la charge de travail des experts** grâce à des analyses opérationnelles simplifiées qui libèrent vos collaborateurs les plus compétents et leur permettent de se consacrer à l'amélioration et l'innovation plutôt que la résolution de problèmes.
- **extraire davantage de connaissances** en collectant et en stockant moins de données. En s'appuyant sur le contrat comportemental de la conception de référence, Apstra sait ce qu'il faut chercher et collecte uniquement ces informations. Ce traitement « à la volée » permet de réduire considérablement les besoins de stockage et de post-traitement des données non intéressantes. Votre infrastructure fonctionne de manière plus efficace et reste compétitive alors que vous réduisez les coûts.
- **détecter les problèmes rapidement et facilement.** Apstra identifie les éléments pertinents exploitables dans une « interface unique » et supprime le bruit des symptômes qui ne sont que les artefacts d'une cause racine identifiée.
- **automatiser les workflows complexes.** Le système Apstra vous permet d'automatiser les workflows de dépannage riches en contexte (qui sont autrement encombrants, inefficaces, chronophages et coûteux).
- **assurer la maintenance sans contact.** Puisqu'il est constamment synchronisé avec l'intention, Apstra offre une maintenance sans contact et sans coût en cas de changement. Par conséquent, il est résilient, répond automatiquement aux changements et économise des coûts importants tout en maintenant les pipelines de traitement des données.
- **se débarrasser du développement interne coûteux**. Intégrer un pipeline de traitement des données en interne est coûteux et fragile, demande du temps et des ressources ainsi qu'un travail important de la part des experts du domaine.
- **fournir des résultats très précis** par rapport aux approches de machine learning/intelligence artificielle.
- **adopter une approche indépendante des fournisseurs**, vous donnant la liberté de choisir parmi les meilleurs fournisseurs et les meilleures capacités.
- **utiliser des API communes** sur toutes vos infrastructures cloud publique/privée/hybride.

Évoluer à partir du jour 1 : étude de cas

Apstra est conçu autour de l'évolution, dès le premier jour. Examinons ce client qui l'a utilisé pour effectuer une validation étendue avec des résultats impressionnants :

- **Infrastructure physique :** 6 000 validations de statut d'interface ; 1 000 validations de câblage ; 36 000 compteurs d'erreur ; 10 000 validations des mesures de puissance, température, tension
- **Plan de données L2/L2** : 12 000 compteurs d'abandon de file d'attente ; intégrité de centaines de MLAG ; 3 000 validations de STP (Spanning Tree Protocol) ; notifications des changements de statut
- **Plan de contrôle** : intégrité de 1 500 sessions BGP ; environ 500 sauts suivants attendus/routes par défaut
- **Plan de capacité** : environ 500 analyses des tendances avec des seuils configurés pour l'utilisation des tables de routage ; tables ARP (Address Resolution Protocol) ; tables multicast par routage et transfert virtuels (VRF) ; 6 000 validations d'utilisation de liaison

- **Conformité** : garantir que toutes les versions de système d'exploitation attendues sont exécutées sur les 100 commutateurs environ
- **Multicast** : 2 500 validations pour les voisins PIM (Physical Interface Module) attendus ; 300 vérifications de point de rendez-vous ; 25 validations concernant la détection de modèles anormaux dans le nombre de sources, de groupes et de paires source-groupe sur les points de rendez-vous

De manière générale, à la place d'une interface unique avec 82 000 entrées, le client obtient une interface simple avec un tableau de bord personnalisé selon ses spécifications qui catégorise les anomalies, qui :

- étaient correctes à 100 % (et non présumées selon les statistiques)
- n'ont nécessité aucune maintenance, puisqu'elles étaient constamment synchronisées avec la topologie et l'intention du client
- fournissaient le contexte exploitable pertinent (l'écart, la nature de cet écart, l'état souhaité)
- étaient enregistrées sur l'espace de stockage et les besoins de traitement (seulement 9 Go de RAM, 96 Go d'espace disque requis)
- évitaient d'écrire des pipelines de traitement des données complexes et fragiles selon des processus internes ou anciens

Conclusion

Ne pas pouvoir apporter de changements fiables dans votre infrastructure informatique freine significativement la croissance et l'innovation. Apstra supprime ce problème et vous débarrasse pour toujours des « infrastructures héritées ». Vous pouvez ainsi effectuer des changements de manière fiable, pour ensuite innover avec constance et rester compétitif.

À propos de Juniper Networks

Juniper Networks simplifie les réseaux avec des produits, des solutions et des services qui connectent le monde. Nos capacités d'innovation nous permettent d'écarter les obstacles et de briser la complexité des réseaux à l'ère du cloud pour éliminer les difficultés que connaissent nos clients et partenaires au quotidien. Pour Juniper Networks, le réseau est un moyen de partager des connaissances et de favoriser un progrès au service de l'humain. Pour cela, nous déployons beaucoup d'efforts pour concevoir des réseaux automatisés, évolutifs et sécurisés, capables d'évoluer au rythme des entreprises.

Siège social et commercial
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089, États-Unis
Téléphone : 888.JUNIPER (888.586.4737)
ou +1.408.7452000
www.juniper.net

Siège EMEA et APAC
Juniper Networks International B.V.
Boeing Avenue 240
1119 PZ Schiphol-Rijk
Amsterdam, Pays-Bas
Téléphone : +31 0 207 125 700

Engineering Simplicity