Distributed Data Repository

Problem

The complexities of large-scale datasets (in the petabytes and growing) exacerbates problems with concentration risk, and QOE suffers. Without reasonably performant distributed access at the edge, and tools to segment access logically and geographically, the pendulum swings back to distributed data sprawl.

Solution

Deploy a single namespace data service that is available in all edge node locations, optimizing for high availability and data protection. It can be immutable to protect against human error and data corruption, and provide policy-based controls to address logical and geographical data segmentation. This is the same proven technology cloud providers use to support massive scale, multitenant data services (i.e., private cloud storage, object storage like S3, etc.). It has not been widely used by enterprises since it needs multiple geographical locations and an optimized WAN to be truly effective. IOA solves that. Geographically place data nodes in each edge node (and cloud environment(s)). From there, built-in algorithms interpret policies and store the actual data in a way that protects from device, location or even regional failures without losing data or access. This offers far more protection than a "copy" and uses much less storage. Data services are also optimized for integration, supporting multiple interfaces (web, APIs, file system, etc.). The technology is not new, it's a data abstraction layer across underlying technologies realized with IOA.

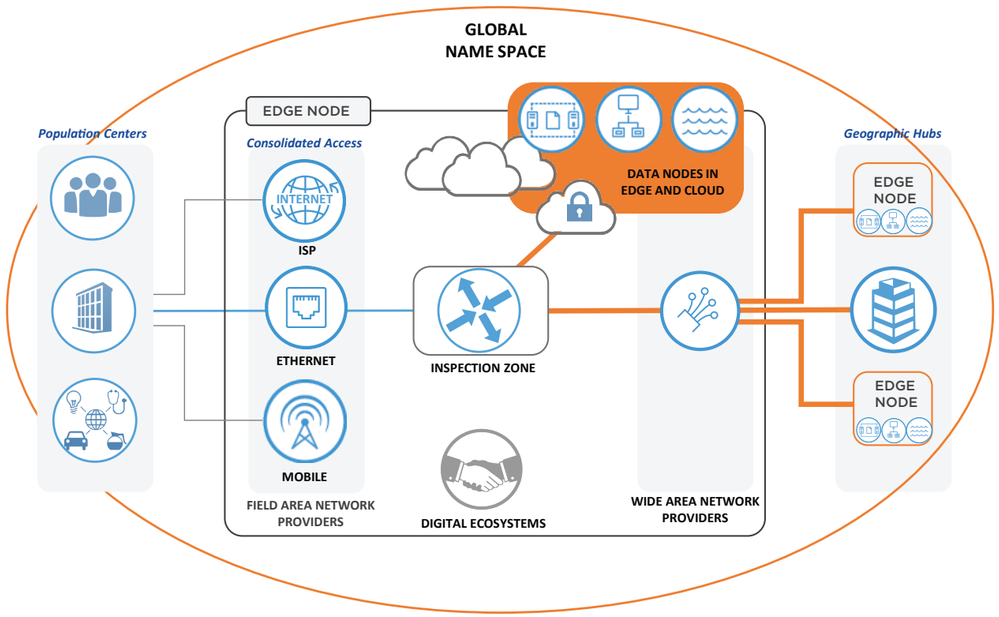
Constraints

1. Centralized data architectures solve for localized data requirements at medium scale (by today's standards), and were not designed for geographically distributed access at a scale well beyond current design limitations.
2. Data sovereignty and privacy regulations force the geographical localization of some data types. These same concerns make cloud as an alternative controversial, or not an option at all.
3. Industry compliance requirements also limit what can be done with data.
4. As businesses and applications struggle to shift to the edge, the management capabilities needed to bring the data are not in place.
5. New data is continuously being generated at the edge, either creating more sprawl, or experiencing a bandwidth backhaul problem with ongoing capacity and its inherent risks.

Steps

1. Design the placement of data nodes. Implementation can vary; however, a best practice is to have a minimum of four locations (cloud environment virtual machines count) and for optimal protection 16 nodes (four locations each with four nodes).
2. Place the first data node and start the service. As more data nodes are added, your private data cloud/namespace will expand in size and the service updates itself.
3. Integrate the service with boundary control, inspection zone(s) (Security Blueprint*), policy enforcement and API management (Application Blueprint*).
4. Apply event processing and monitoring.
5. Establish logical capacity buckets and access groups with protection and placement policies.
6. Continue to add more data nodes to scale namespace capacity. Data nodes deployed in the cloud can be backed by cloud-provided storage (e.g., S3). Policy determines cloud use.

Reference View



Forces

- Data is growing — from GBs, to TBs, to PBs and now approaching ZBs. Proportionally more data is being generated each year than what was already stored.
- Data is no longer centralized. As traffic and processing have shifted to the edge, data growth and consumption are being drawn with it. The physics of latency along with ever-increasing bandwidth costs mean that backhauling all this data is not sustainable.
- Data is becoming more valuable — to the business, to customers, to competitors and threats. Businesses that were product driven are now going to be data driven.
- This drives the need for data services that provide data availability at the edge and accommodate logical and geographical data segmentation.

Results

- A distributed service with centralized management solves two previous constraints.
- Designed to handle zettabyte scale data sizes.
- Accessibility reduces the volume of data moved and copied—and therefore stored (disk space) significantly, which reduces WAN bandwidth and storage costs.
- All data can be automatically encrypted at the data layer (the key is also dispersed and not stored in any one place).
- With security and policy management integration, ensure compliance consistency in all regions.
- Any non-latency-sensitive data requirements use the repository as primary storage and archive ("eventually consistent" regionally).
- Examples of use include shared drive, package distribution for apps/containers, logging repository, staging area for analytics, etc.

* Security and Application Blueprints — IOAKB.com