

Summary

A scalable micro-services architecture delivers autonomy, agility, and data quality.

- Timeline : 6 months from clean sheet to 35 domains
- Team : One senior architect plus a modeler and API developer
- Impact: Transformation plan for 200+ systems & services

- Architecture: Domain driven RESTful Microservice API design.
- Relationships: Engaged & sold ideas to 20+ business areas.
- Technology : Prototype deployments to AWS API-GW & Kafka

The Challenge

To develop a modern, scalable integration architecture for a large & complex organisation

The Home Affairs portfolio is a very large organisation that includes immigration, customs, policing, and intelligence functions. The current state system integration framework presents some challenges.

- Over 1000 distinct integrations between 200+ IT systems used by 30,000+ users.
- Tight coupling between systems imposes very complex dependencies and severely limits agility.

- Database change data capture processes impose complex “deciphering” ETL work on consumer systems.

In recognition of the challenges of scaling such a complex and tightly coupled framework, the portfolio released a 2020 technology strategy that demanded both autonomy / agility and high quality centralised intelligence information for border risk decision making. A new integration strategy was identified as key to achieving both goals.

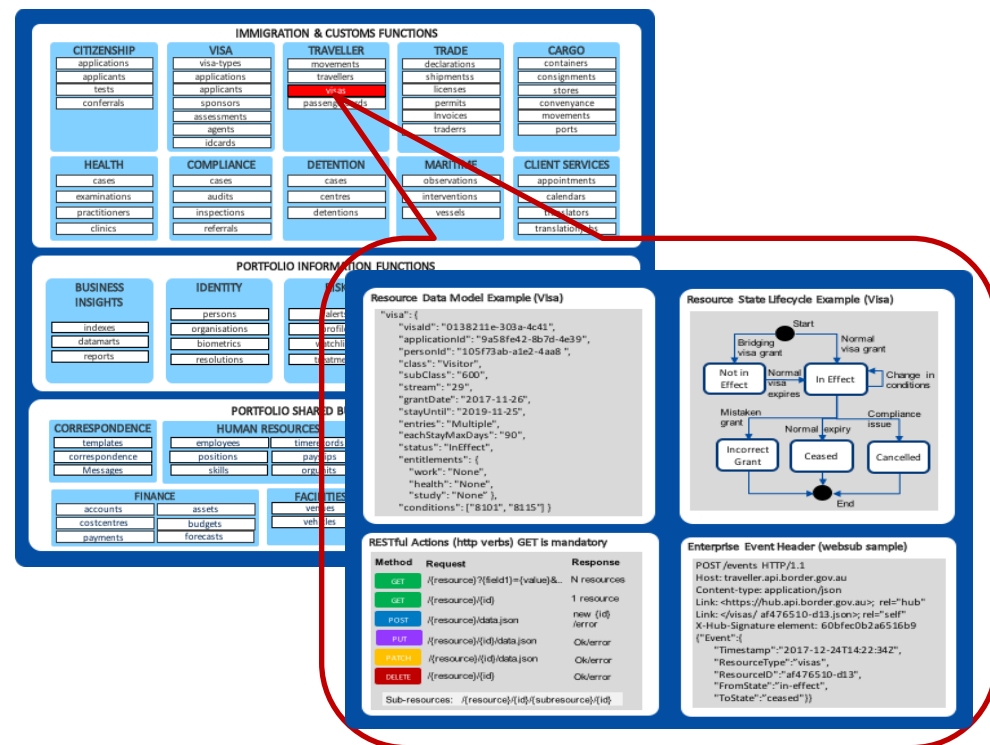
Our Approach

Simplify & decouple through domain driven design, RESTful micro-services & events.

The previous Home Affairs integration model was based on a centralised ESB-centric service design and implementation function. Most integrations were based on ad-hoc project needs defined late in each project lifecycle. The result was a patchwork of tightly-coupled systems where each service was designed for the specific needs of one consumer with little reuse.

With system integration previously considered as a purely technical activity, the biggest challenge was to drive a change in thinking across a large number of IT system teams and their executive. Instead of “how shall we connect A to B for this project?” the thinking needed to be more like “what does our business unit do and how should we expose our services for self-service consumption by any other business unit?”.

The critical success factor was to develop a top-down domain & resource architecture for the portfolio that defined the required microservice granularity. Then each domain was empowered to design & implement their RESTful micro-services & events within the centrally governed high level domain architecture.



Key Results

30 Autonomous business domains linked by simple APIs & events.

Integration Architecture documented

- As a detailed document and as a compelling 1-page poster – which is appearing on walls throughout the portfolio.

Approved by executive and embedded in capital plan

- A high level why-what-how slide deck was developed to drive executive support – which led to senior approval and capital funding.

Accepted by business domains and IT system owners

- Real examples and “show the thing” prototypes were used successfully to engage business domains and IT system owners.

Technology platforms including API gateway & enterprise event hub designed and funded.

- Design and prototypes of shared technology platforms (API gateway and event hub) were delivered.

Capability building program implemented.

- The decentralised delivery model for microservice design & implementation required a capability building program to empower IT teams.

Complexity reduced and autonomy enhanced

- 700 ESB transformations reduced to 16 resources in 5 domains.

Technologies

Successful early prototypes of API gateway and event bus

Standards

Principles

Domain Driven Design
RESTful Architecture
Event Driven Architecture

Technologies