



Implementing an API Platform

lessons learned from real life customers

Axel Grosse
@apiaxl

#APIDaysNZ

07 October 2016



axway

SPARK 2016

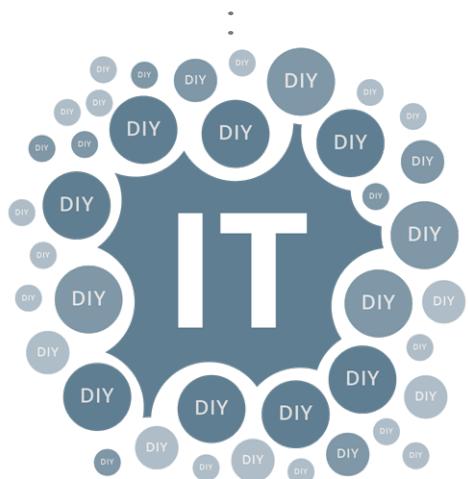
NOVEMBER 3, 2016 8:00 AM US PST // **REGISTER NOW AT AXWAY.COM/SPARK**



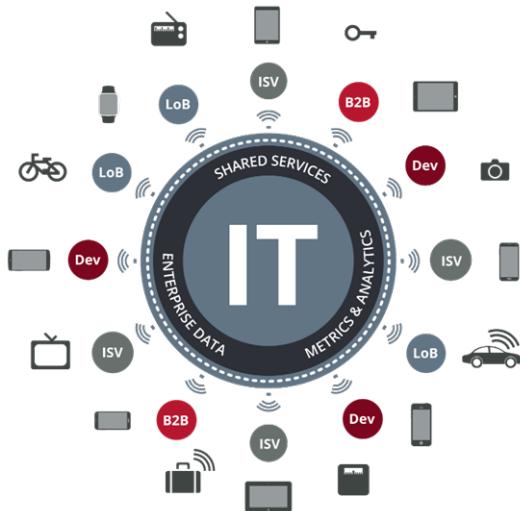
The Context

IT is an essential part of a companies digital journey

PRESENT



FUTURE:



DIGITAL

“SCIENCE PROJECTS”

@apiaxl

DIGITAL

BUSINESS ENABLEMENT

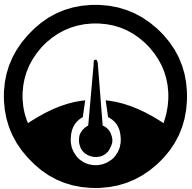
#AxwayAPIDays

The digital vision

the business and the architects dream



The API Platform



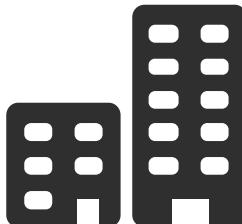
Performance



Simple to use



Secure



Enterprise



Experience



Visibility



Always-On



Designed
for change

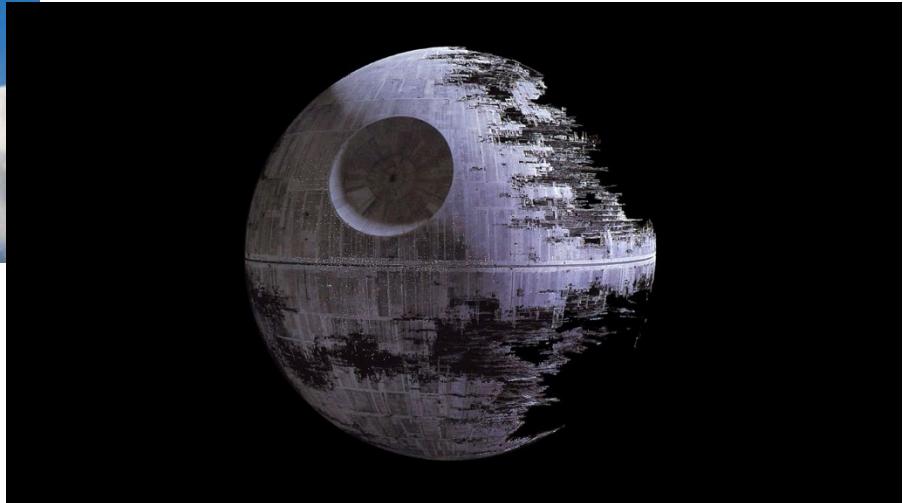
Vision meets Reality

There is never a green field



Funding challenges
architecture best practices

have to exists in the given & planned
DC architecture and processes

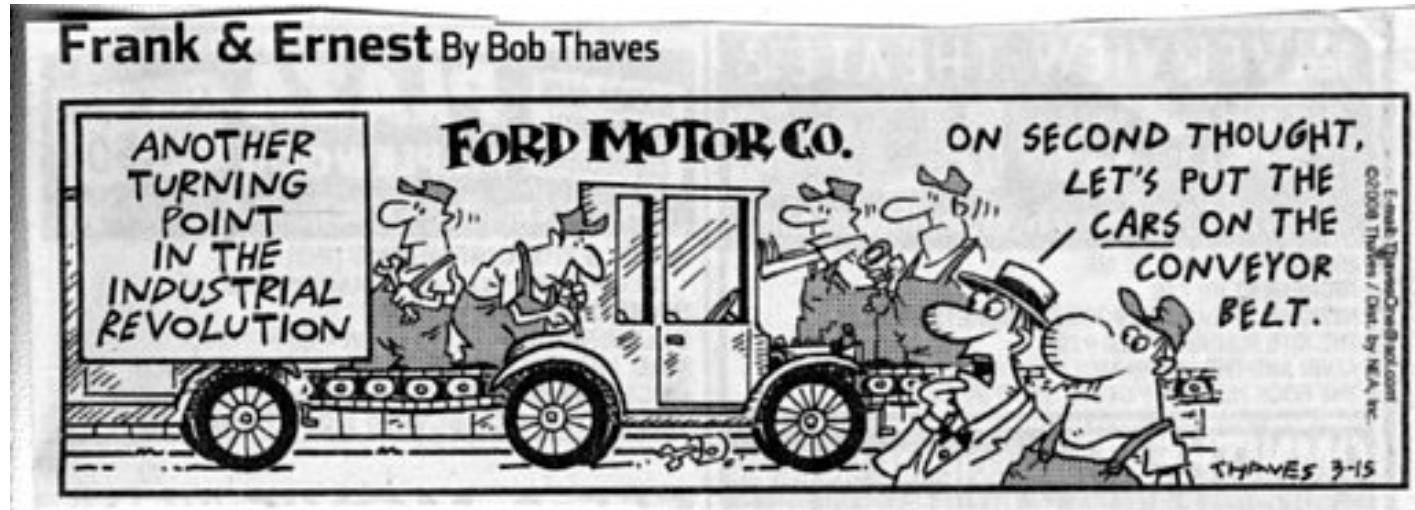




The API Platform Journey

Standards and Guidelines

Provide and re-enforce



- need to be developed and designed upfront
- Need to be constantly re-enforced
- Must be ready for evolution

Standards and Guidelines

Architecture Standards

Multi-layer Services

- Internet facing APIs perform to few days)
 - Validation and Transform
 - Orchestration of underlying prior to orchestration
 - Usage and Access Control authentication, authorization
 - APIs sometimes reused, in some cases just specific
- Business services (*Time*)
 - Pragmatic in design (More granular than there predefined)
 - Abstract backend system
 - Access and Usage control

API Specification Standards

- JSON centric and RESTful
- All APIs Richardson's Maturity standard for API design with
- Successfully implemented RESTful insight which necessarily means
 - Everything is addressable
 - Correct usage of HTTP

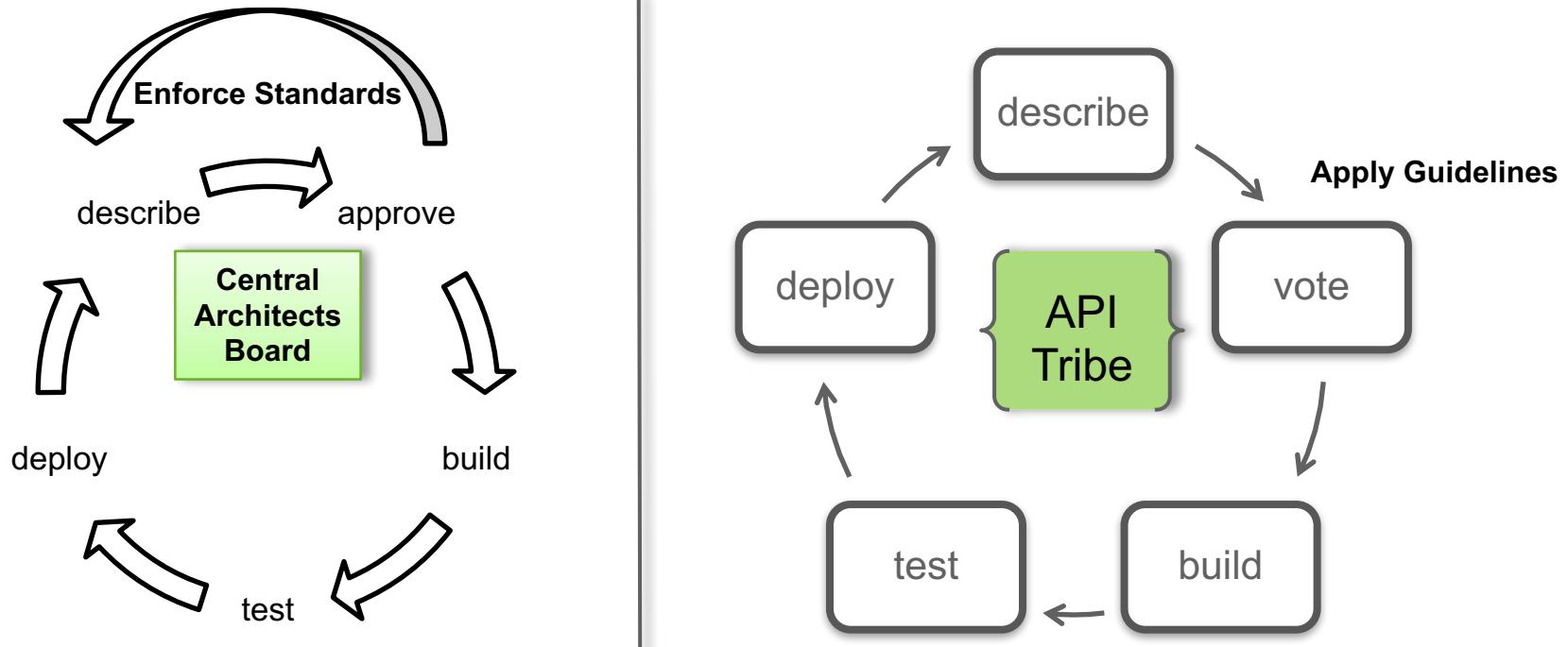
Authentication Standards

- OAuth 2 – Resource Owner password credential flow for Mobile App and Kiosk originating API calls
- OAuth 2 – 3 Legged flow for websites
- Axway API gateway also provides Single Sign-on across all sites

90% of API consumers are self-developed apps/system with predefined hard coded predictable API consumption logic hence it was decided to not pursue HATEOS.

- API definitions via Swagger
- API Catalog used as a Portal for integration developer to explore and learn APIs

API design governance ... open debate



- **Integration Developers not used to UX driven API's**

- Training
- Configure few display API
- Web Developer to join the team

- **Test Driven Development is desired but is mostly not the actual culture**

- In absence of TDD identify high priority tests
- For Automation use easy to use and familiar tools like JMeter
- Once test cases are automated integrate them with a Jenkins Build pipeline



- **Continuous Deployment Infrastructure not ready ... common challenge**
 - CD is moved to the platform phase
- **3rd Party systems integration into internal landscape via APIs**
 - Making it part of the API Platform enables Usage & Quota Control
- **Security joins late in the journey**
 - API Platform Design for evolution helps to react
- **Absence of a Single Sign-On system for omni-channel APPs**
 - OAuth2.0 (3 legged) flow was used to implement single sign in across multiple sites



The way to production

If you cannot start in full .. Start iterating

- Running major tests as part of the development lifecycle
- Start small with a dedicated set of API's supporting one application channel
- on-boarding of new projects with platform in mind
- Once live start looking at the KPI's to prove success

What does success looks like in API World?

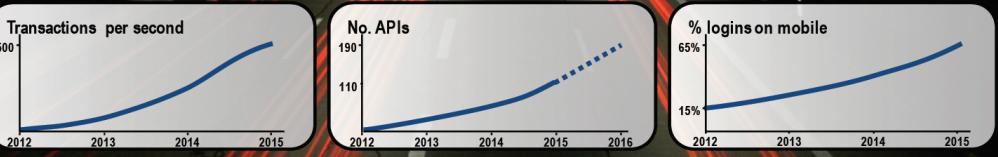


Outcomes

Cost Reduction
Cost per integration: ↓98%
Cost per client UI: ↓85%

Schedule Reduction
Test cycles: **9 weeks** down to **3 weeks** (or less)
Release cycles: **3 months** down to **1 month** (or less)
Ability to perform **12 week experiments**

New Opportunities
Extended use of **cloud** infrastructure
Ability to **partner** more easily
3rd Party developer ecosystem



Year	Transactions per second	No. APIs	% logins on mobile
2012	100	110	15%
2013	200	130	20%
2014	300	150	25%
2015	500	190	35%



"Axway API Management is mission-critical for The Star Entertainment Group. The Axway solution is an incremental revenue generator for us. It displays real-time content across all our channels, and that's what enables customers to engage with us any time."







About Axel

Axel is responsible for Axway's API Management solution across Asia Pacific.

Through his career of close to 5 years with Axway, Axel has worked with many of the region's leading organizations and financial institutions, particularly in Australia, New Zealand and Southeast Asia, and has overseen major implementations with customers as they embark on their digital transformation journeys.

Prior to Axway, Axel worked in Germany with Vordel for 2 years and was the chief engineer driving the API Gateway adoption in the market before Axway's acquisition in 2012.

Axel has also spent 10 years at Oracle, covering the Middleware and Integration stack.

