redhat.

MONOLITHS TO MICROSERVICES: APP TRANSFORMATION

Hands-on Technical Workshop

Thomas Qvarnström
Sr. Technical Marketing Manager
Middleware BU

James Falkner
Sr. Technical Marketing Manager
Middleware BU

API-LED MODERNIZATION

redhat.

WHY APIs

More than a technology, it's a way of doing business

- Your product is your business, both internal and external
- Entrepreneurial approach to solving problems and leveraging opportunities
- Customer service driven behaviors
- Support overarching organizational strategies with data-driven decisions

redhat.

THERE IS NO “OFF” BUTTON FOR LEGACY APPLICATIONS

redhat.

> “When designing a new application you should design it in such a way as to make it easier for it to be strangled in the future. Let's face it, all we are doing is writing tomorrow's legacy software today. By making it easy to be strangled in the future, you are enabling the graceful fading away of today's work.”
>
> - Martin Fowler

redhat.

APPLICATION MODERNIZATION

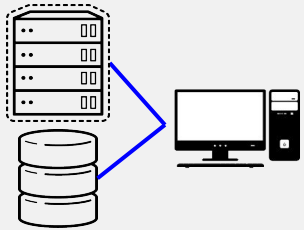
Adopting legacy applications to deliver value for modern business needs

redhat.

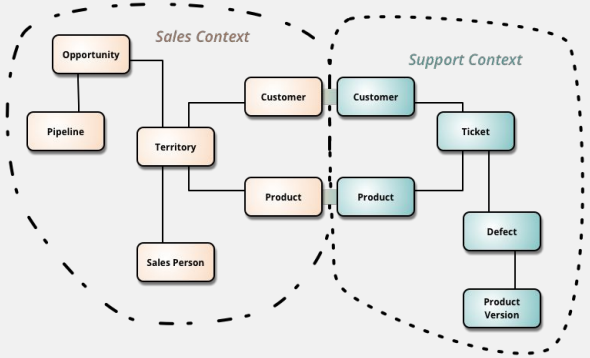
MODERN SOFTWARE ENGINEERING

Methodical Approach to Evolve Systems

MAINFRAME



MONOLITH

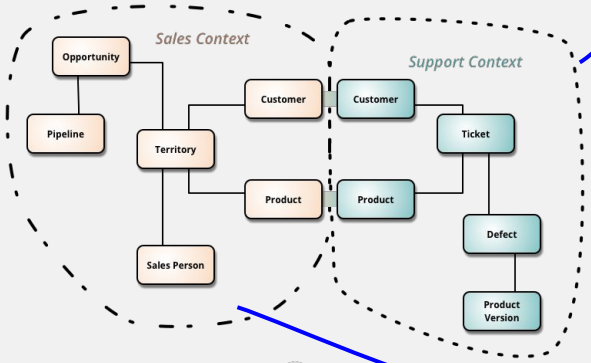


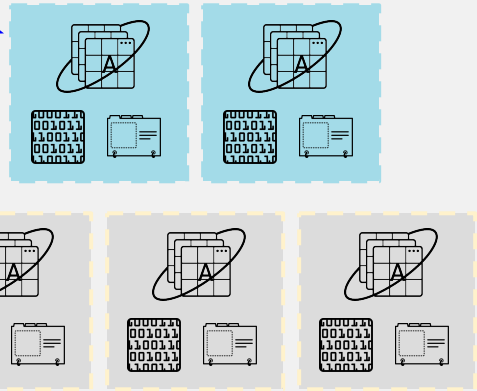
BOUNDED CONTEXTS

redhat.

MODERN SOFTWARE ENGINEERING

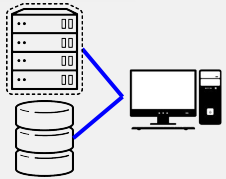
BOUNDED CONTEXTS





MICROSERVICES & APIs

MAINFRAME



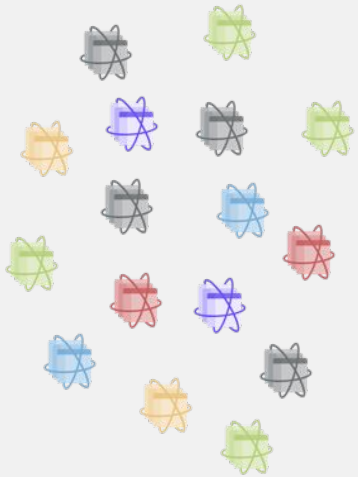
MONOLITH

redhat.

PURPOSE-BUILT TOOLS

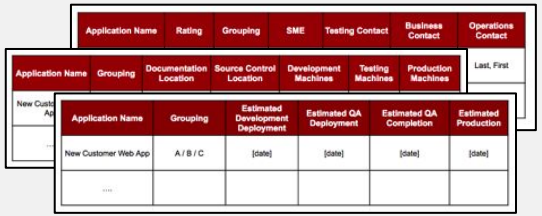
Eliminate guesswork for predictable results

ANALYZE

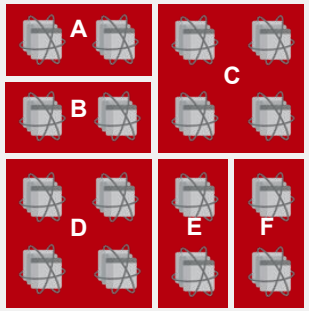


RATIONALIZE & CATALOG

RED HAT®
APPLICATION
MIGRATION TOOLKIT



PLAN



redhat.

API - LED MODERNIZATION IN PRACTICE:

CONNECT AND EXTEND

redhat.

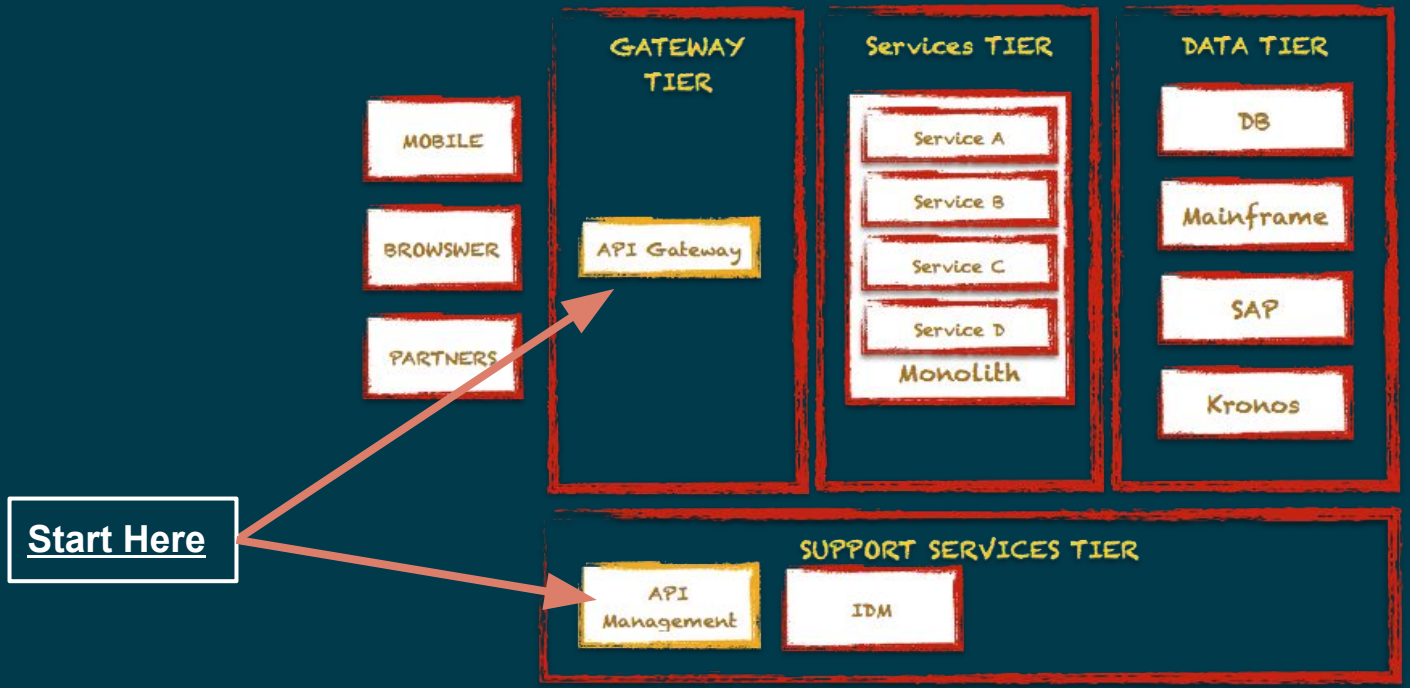
AKA:

STRANGLING THE MONOLITH

redhat.

PHASE 1: STRANGLING THE MONOLITH

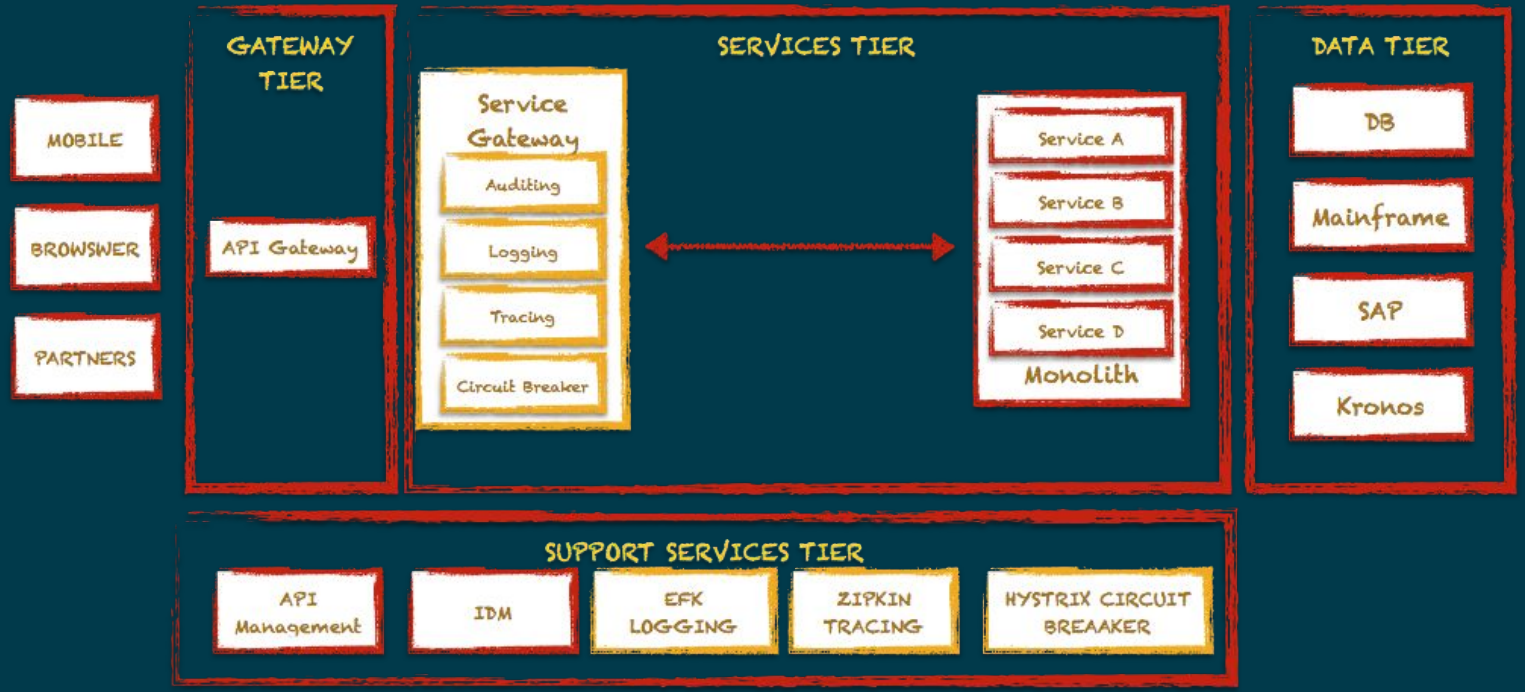
API First



redhat.

PHASE 2: STRANGLING THE MONOLITH

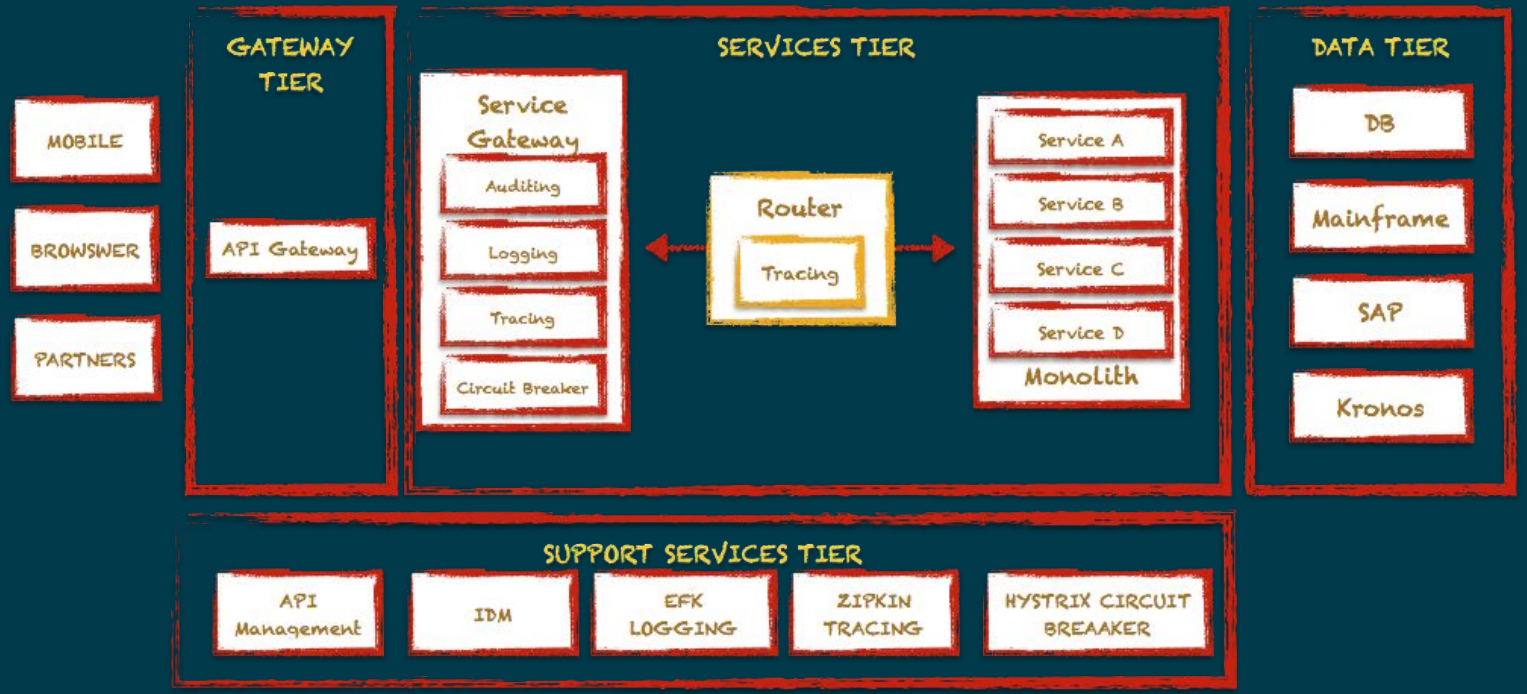
Connect and extend with a Services Gateway



redhat.

PHASE 2: STRANGLING THE MONOLITH

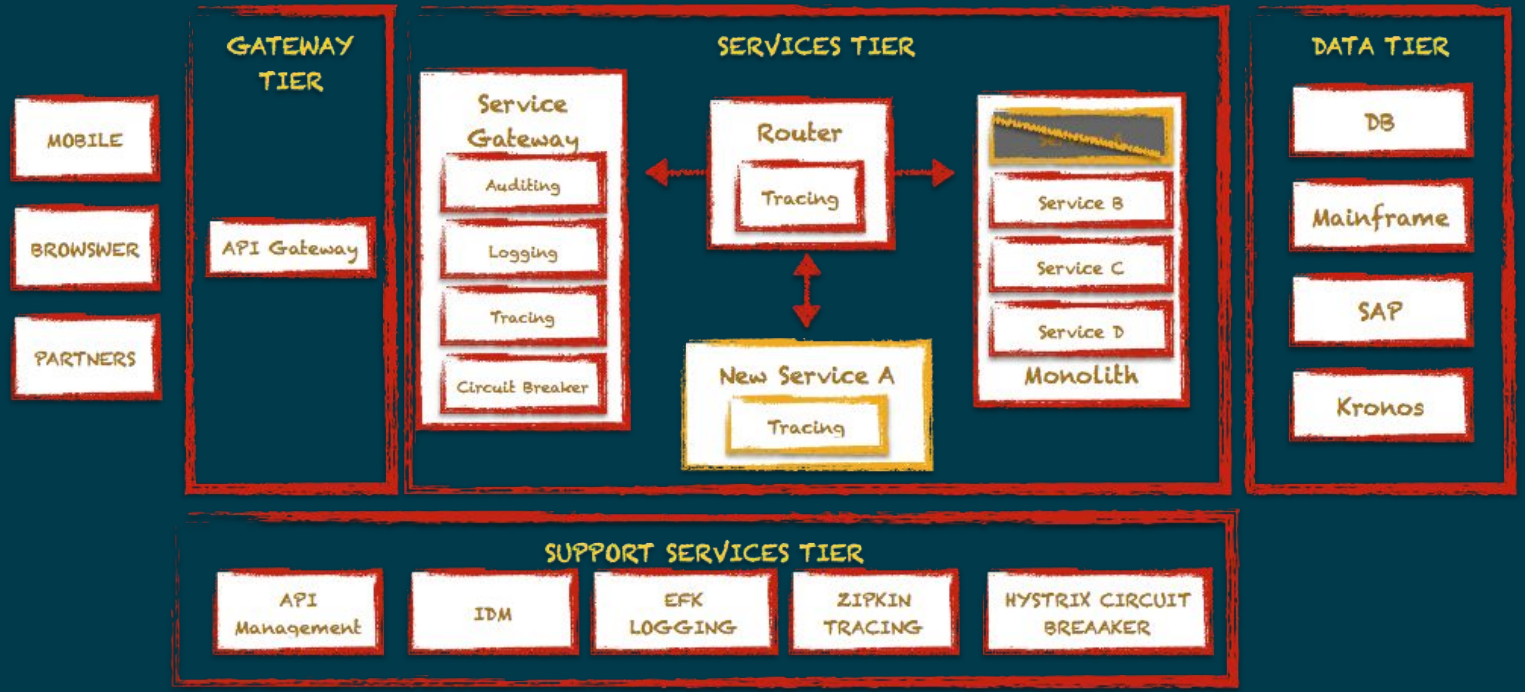
Create a path for new services



redhat.

PHASE 2: STRANGLING THE MONOLITH

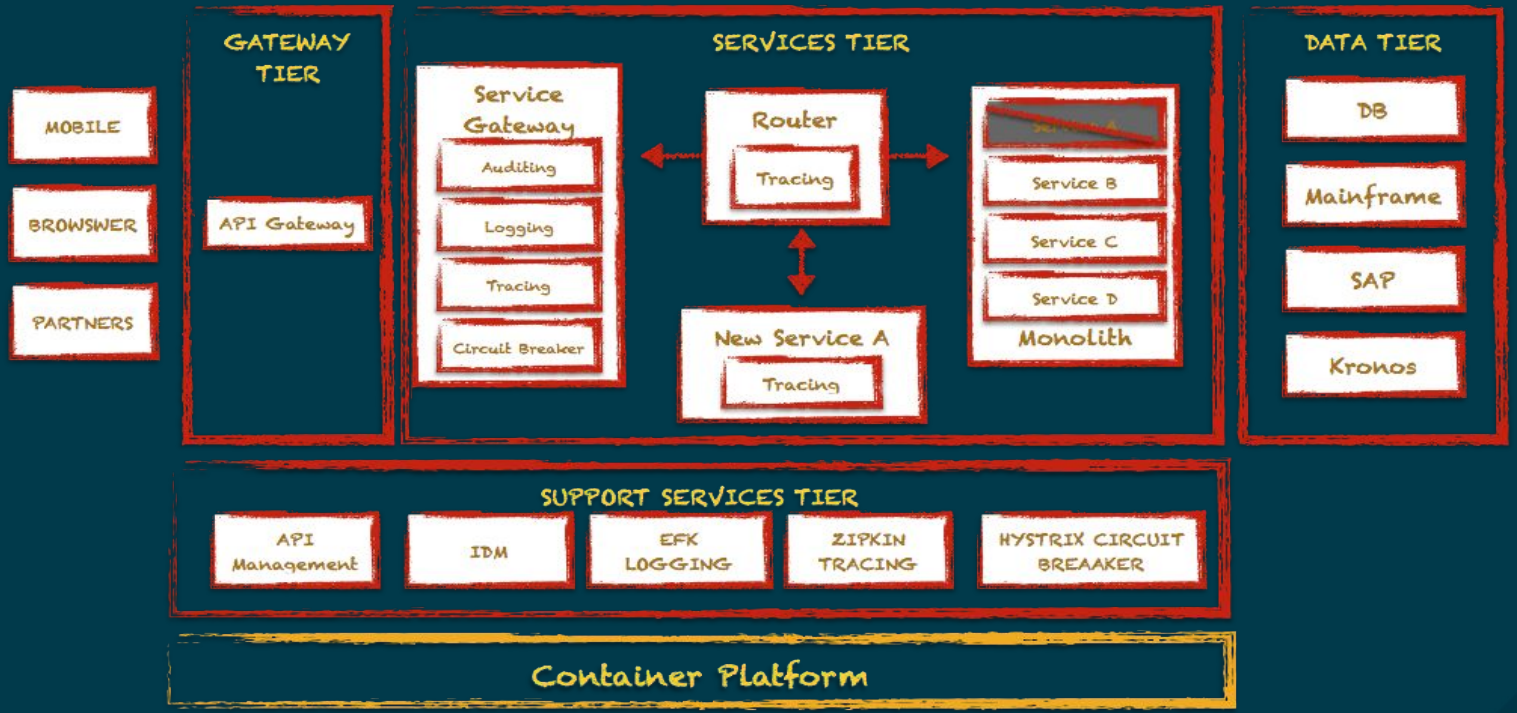
Out with the old, in with the new



redhat.

PHASE 3: STRANGLING THE MONOLITH

Containerization



redhat.

DRILL DOWN

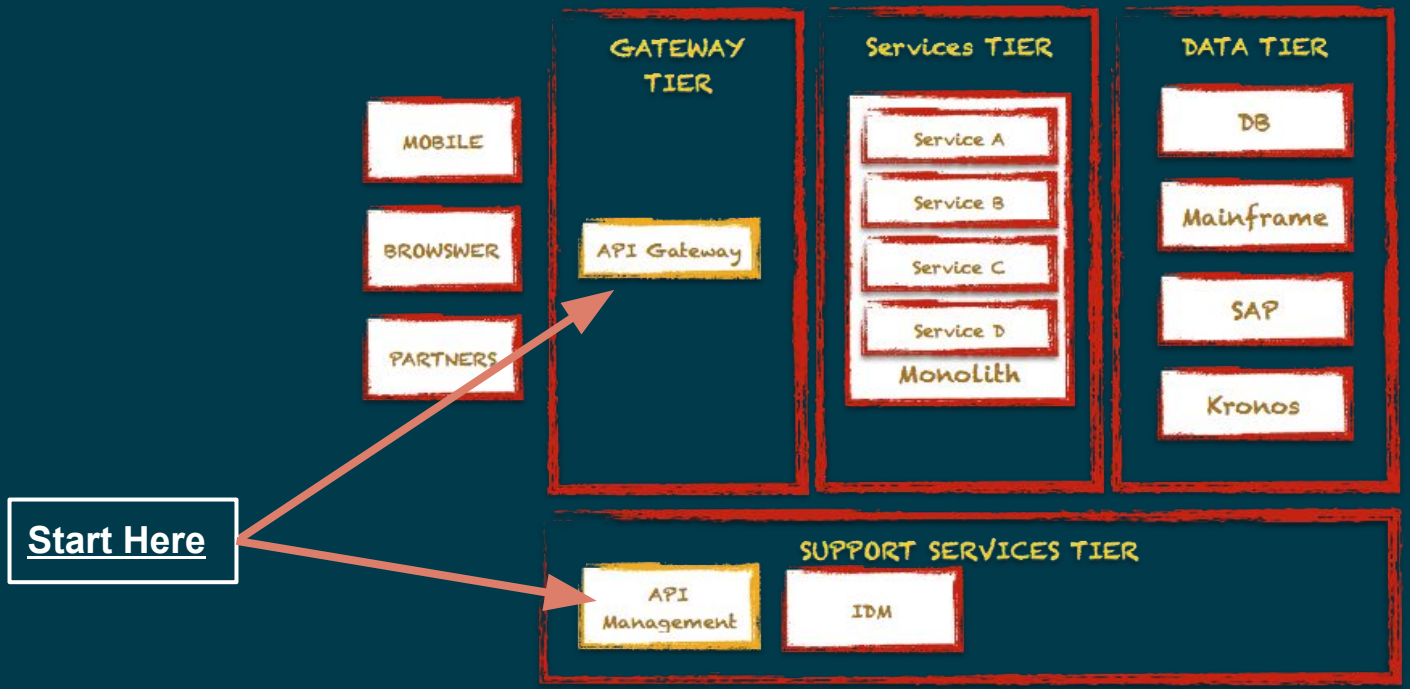
PHASE 1: STRANGLING THE MONOLITH

EXPOSING APIs

redhat.

PHASE 1: STRANGLING THE MONOLITH

Expose APIs



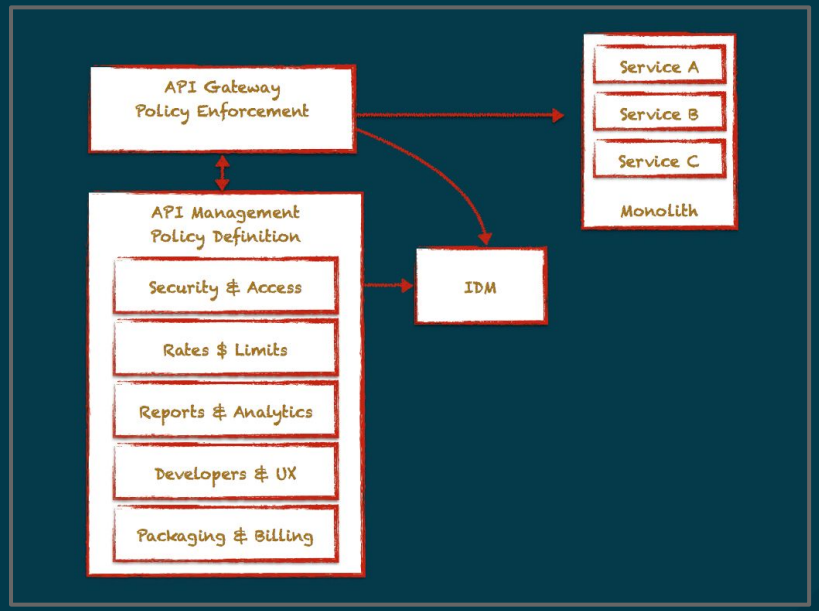
redhat.

WHY API MANAGEMENT?

API Management is where we start growing and learning

5 Components to API Management

- Security & Access Control
- API Contracts, Throttling & Rate Limits
- Reports & Analytics
- Developer & Partner UX
- Packaging, Billing & Payments

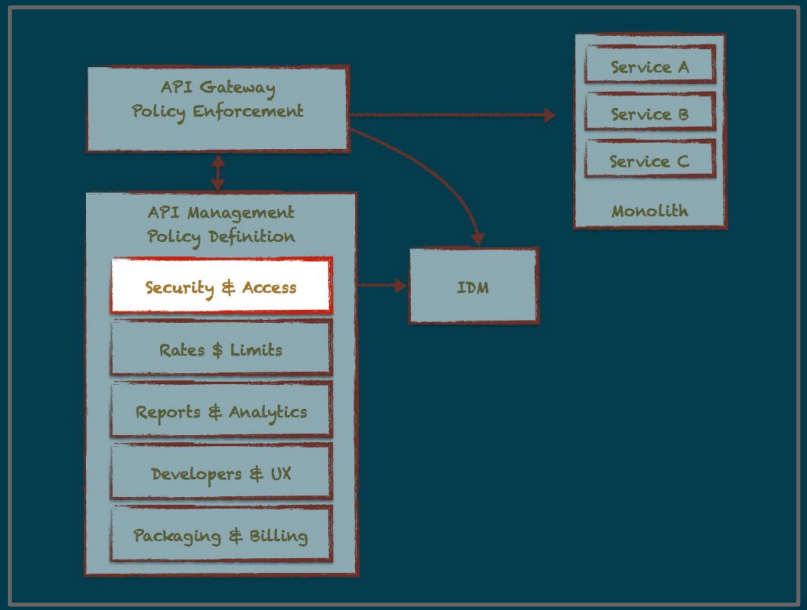


redhat.

WHY API MANAGEMENT

API Security & Access Control

- Authenticate traffic
- Restrict by policy
- Drop unwelcome calls
- Protect backend services
- Generate overage alerts
- Impose rate limits

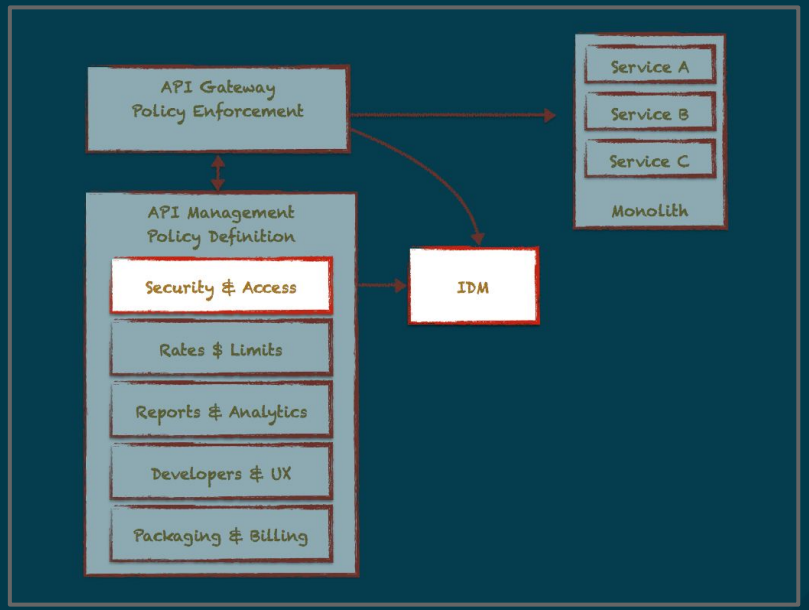


redhat.

WHY API MANAGEMENT

API Security & Access Control

- Multiple authentication mechanisms
  - API Key
  - APP ID / API Key
  - OAuth 2.0

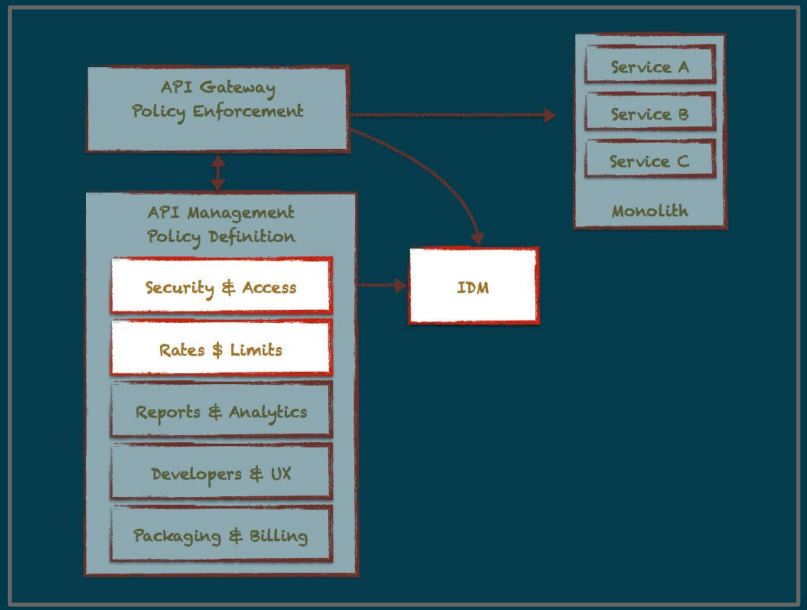


redhat.

WHY API MANAGEMENT

API Contracts, Throttling & Rate Limits

- Allow/restrict access to your API endpoints along with rate limits
- Rate-limit accounts
- User and endpoint level

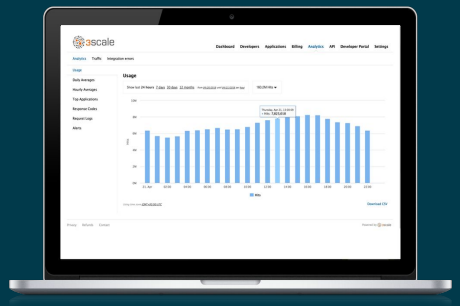


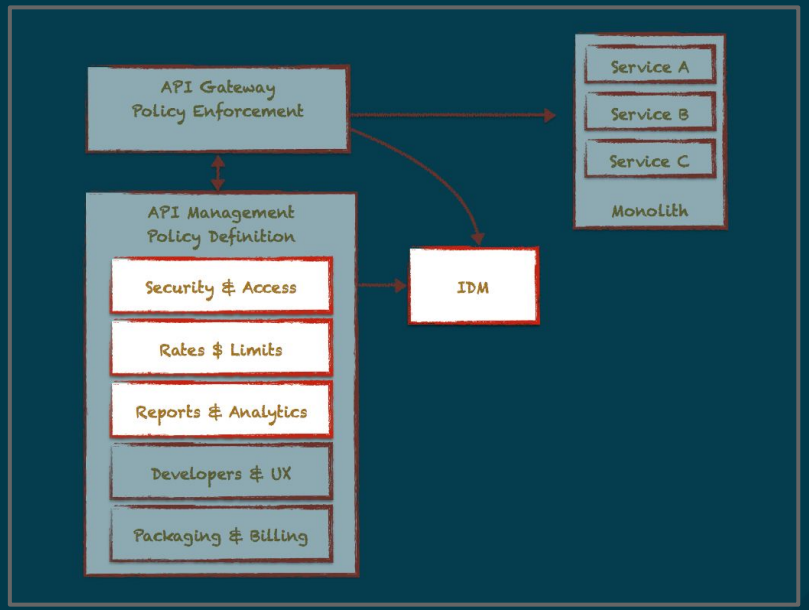
redhat.

WHY API MANAGEMENT

Reports & Analytics

- Analytics providing intelligence about the performance, and traffic patterns
- Visibility who is doing what where
- Automatic based on behaviors



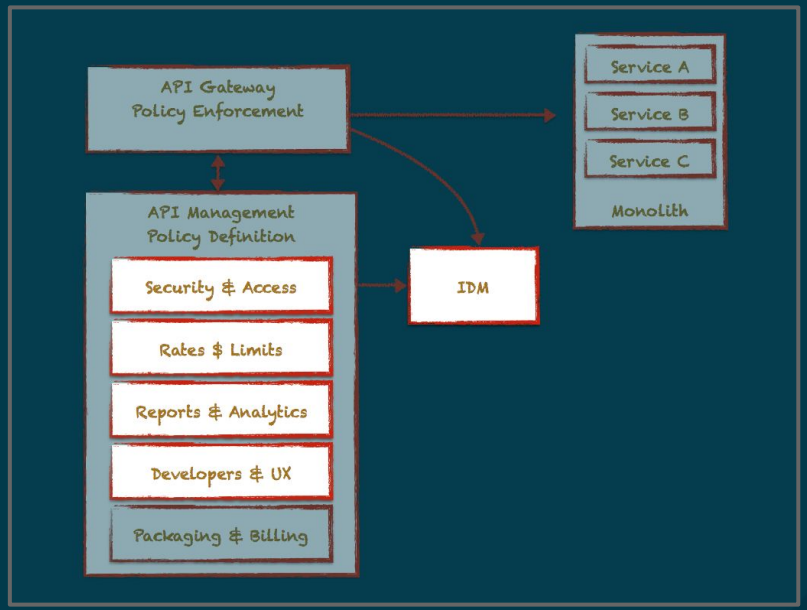


redhat.

WHY API MANAGEMENT

DEVELOPER PORTAL & USER EXPERIENCE

- Your brand
- Your user experience
- Your user interface

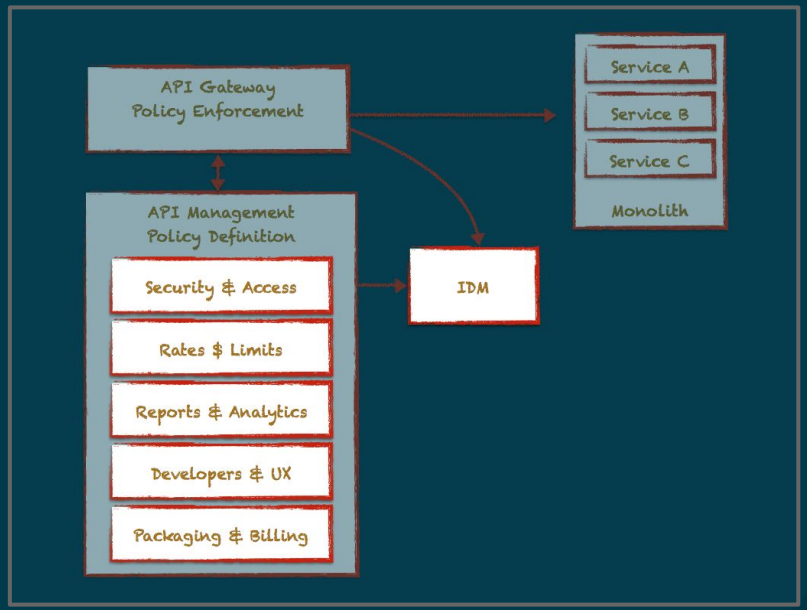


redhat.

WHY API MANAGEMENT

Packaging, Billing & Payments

- Multiple pricing rules
  - One time payment
  - Fixed/Variable recurring fee
  - Cost per unit
  - Tiered pricing
- Billing cycles
- NO CC DATA STORED



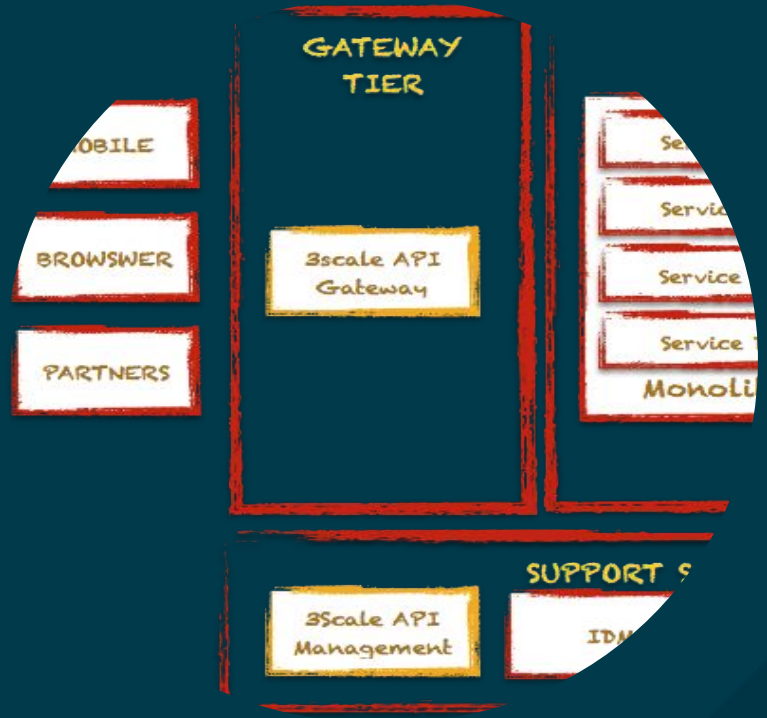
redhat.

3SCALE API MANAGEMENT

Expose APIs Faster with 3scale

Keys Differentiators

- Hybrid Cloud Architecture
- Separation of Concerns
  - Slim DMZ deployments
  - NO DATA AT REST
- 1st class citizen in an automated DevOps tool chain



redhat.

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos