

AN API-FOCUSED APPROACH FOR AGILE INTEGRATION

Christina Lin - Technology Evangelist

Gary Gaughan - Product Manager

Red Hat

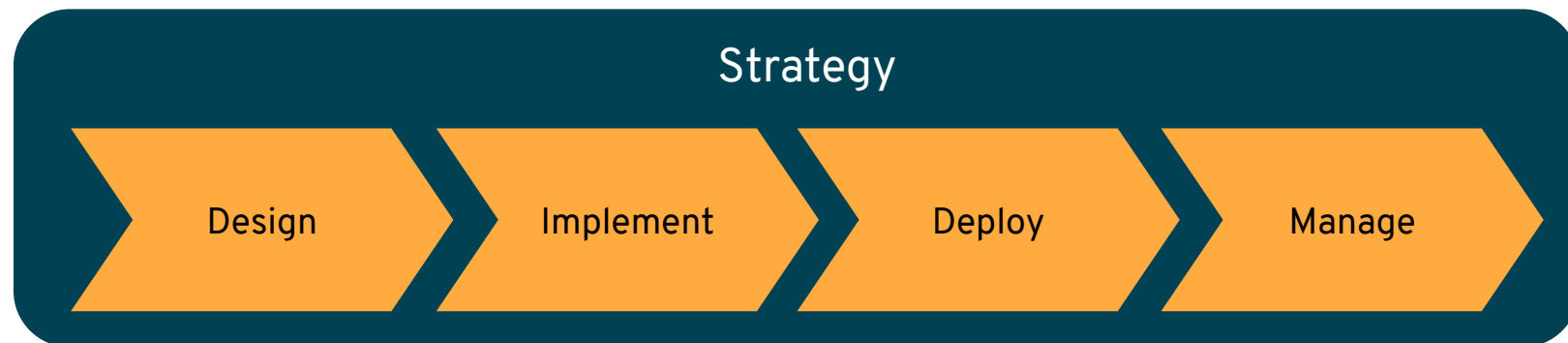
THREE THINGS

Avoid accidental API architectures

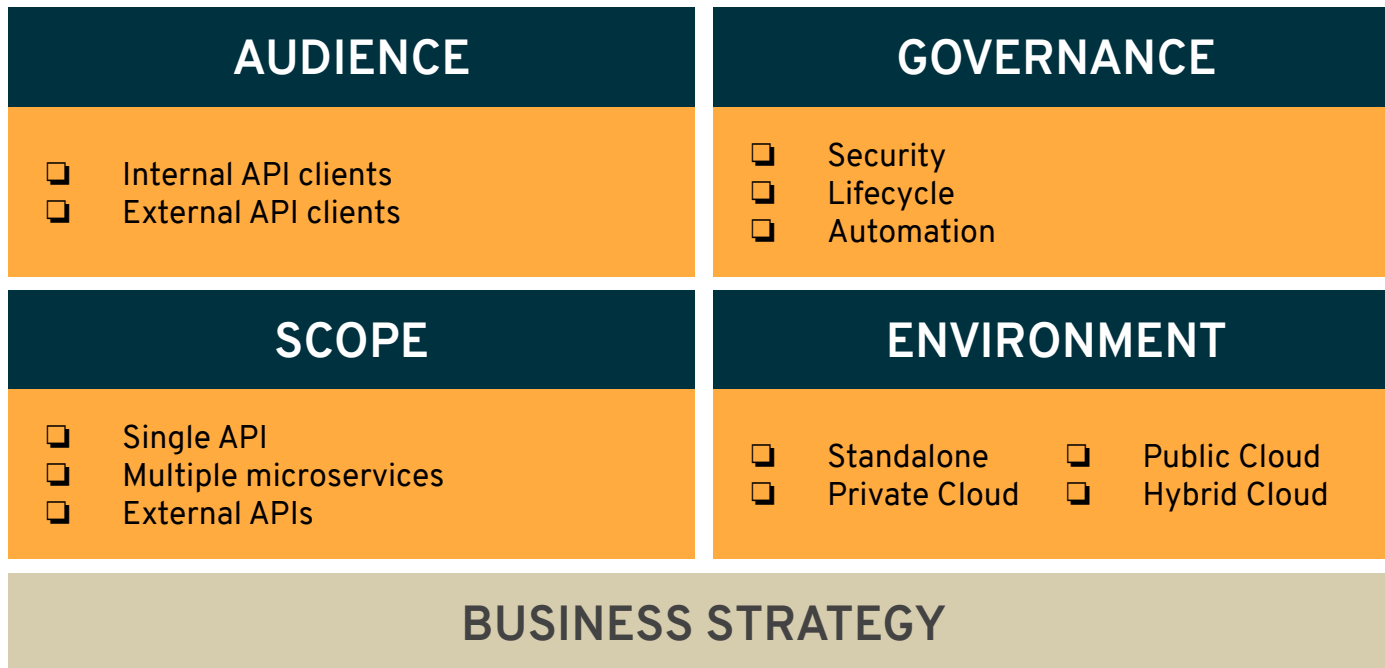
Understand the role of integration in API-centric architectures

Get “hands on” with Red Hat Integration

API-CENTRIC INTEGRATION LIFECYCLE

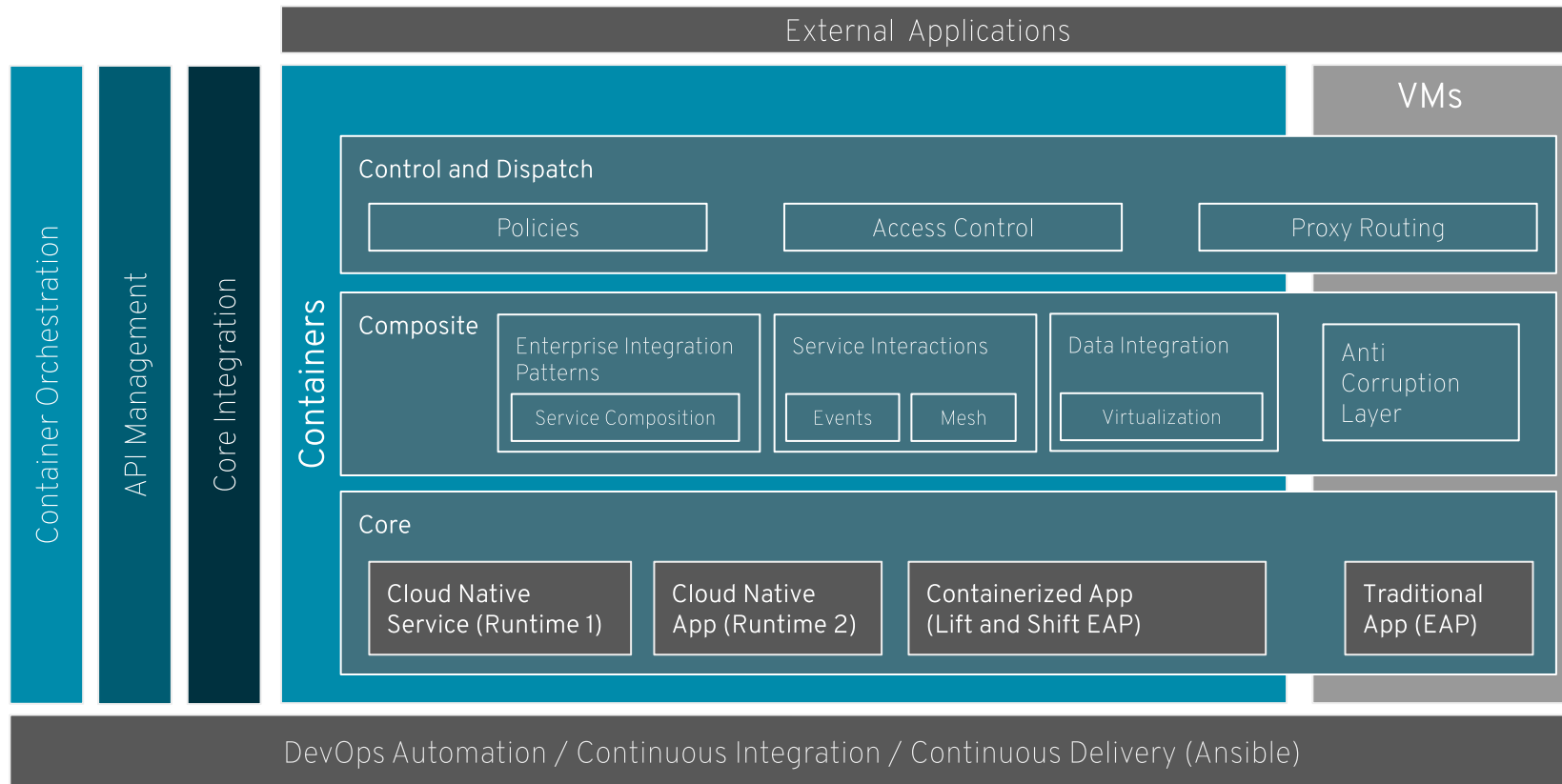


API STRATEGY



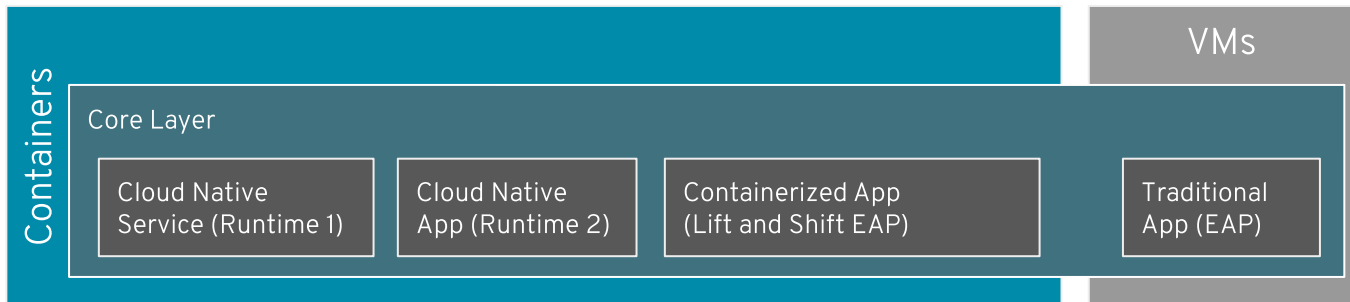
AGILE INTEGRATION ARCHITECTURE

CONFIDENTIAL Designator



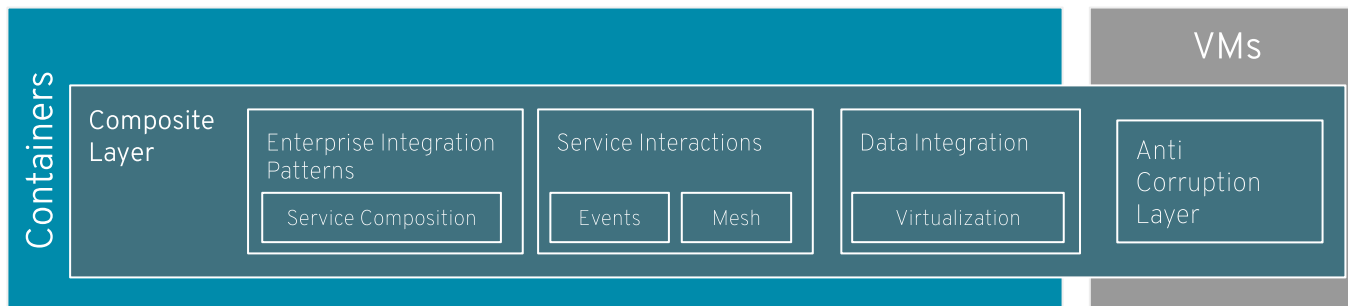
CORE SERVICES

- * Brownfield and greenfield services
- * Microservices and monoliths
- * Delivered independently
- * Independent data contexts
- * Mixed connectivity



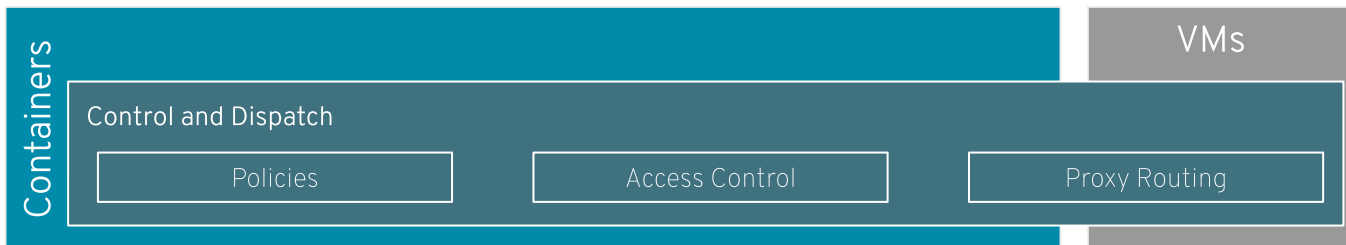
COMPOSITE

- * Service composition
- * Routing and orchestration
- * Data transformation
- * Connectivity
- * API ↔ Event Bridging
- * Legacy facade (ACL)



CONTROL and DISPATCH

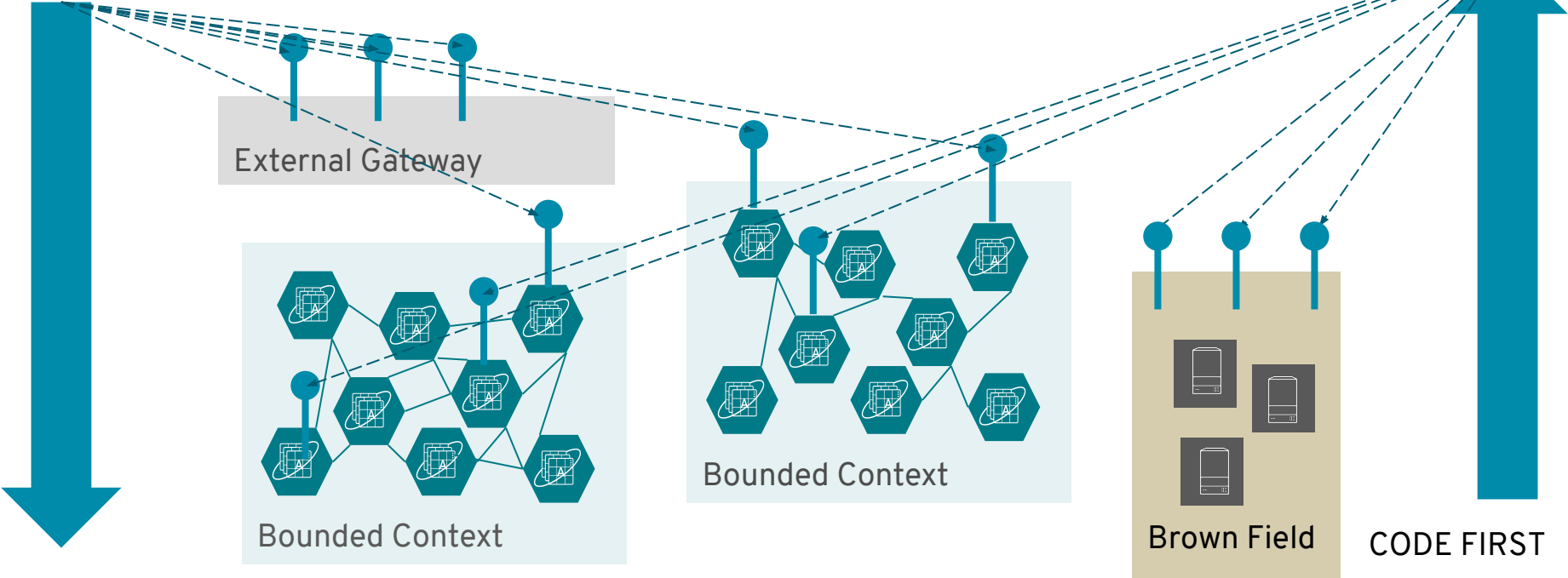
- * Gateway layer for services
- * Access and policy control
- * Developer onboarding
- * Analytics
- * Monetization
- * Scales independent of other layers



API CENTRIC

DEVELOPMENT

CONTRACT-FIRST





CLIENT-FOCUSED

- Design with the API client in mind
- Design with tooling fit for purpose
- Collaborate ASAP

VALIDATE EARLY

- Use API mocking for early feedback
- Skeleton implementation can be just as good as a mock

FAVOR INTEROPERABILITY

- Create API definitions based on standards in open communities
- Maximize tool portability and client generation

DEMO TIME!





HONOR THE TRUTH

- ❑ API Definition is the source of truth
- ❑ Favor generation over translation

NOT ALL APIs ARE THE SAME

- ❑ Standalone
- ❑ Data API
- ❑ Orchestration
- ❑ Event Bridge
- ❑ Legacy Facade

WHICH PERSONA?

- ❑ Developer
- ❑ Non-developer

DEMO TIME!





CONTAINERS

- ❑ Best way to develop services (polyglot, portability, availability, service wiring, advanced deployment, ...)
- ❑ Maximize inner vs. outer architecture pattern

HYBRID ENVIRONMENT

- ❑ Support integration and management of APIs living outside containerized environment
- ❑ Consistent architecture across private, public, and managed cloud

AUTOMATE

- ❑ API-driven infrastructure services
- ❑ Ability to automate application and infrastructure services in a single pipeline

DEMO TIME!





CONTROL

- ❑ Securing APIs
- ❑ Traffic flow control via policy
- ❑ Policy extensibility

VISIBILITY

- ❑ Developer onboarding and engagement
- ❑ Traffic and policy alerts
- ❑ Use analytics to understand how APIs are tracking against business objectives

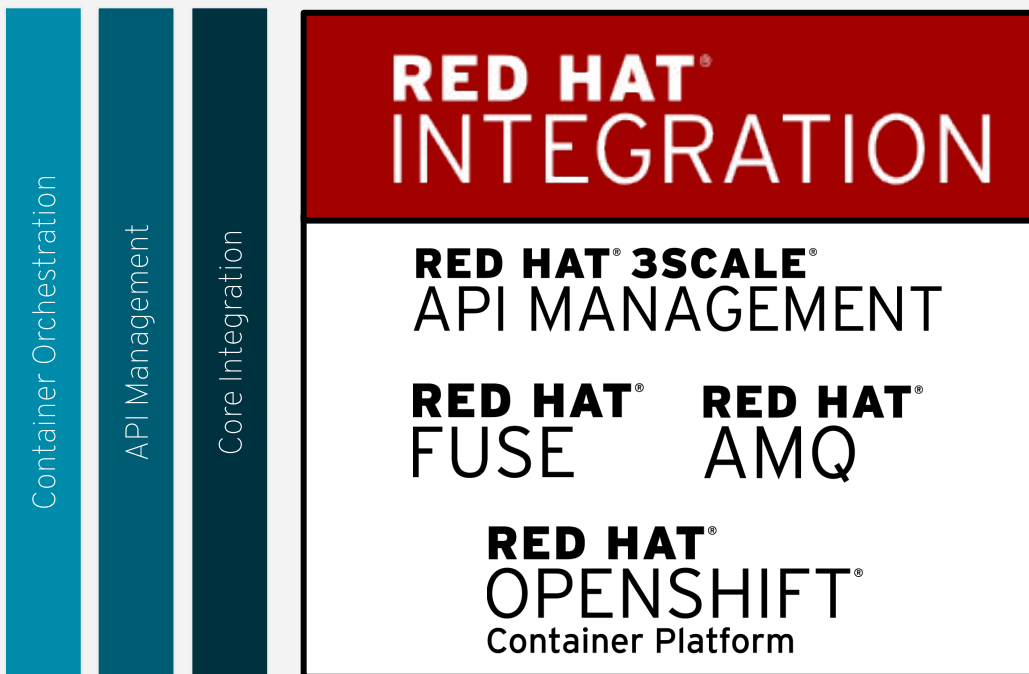
FLEXIBILITY

- ❑ Centralized management and distributed enforcement
- ❑ API management architecture must span multiple environments

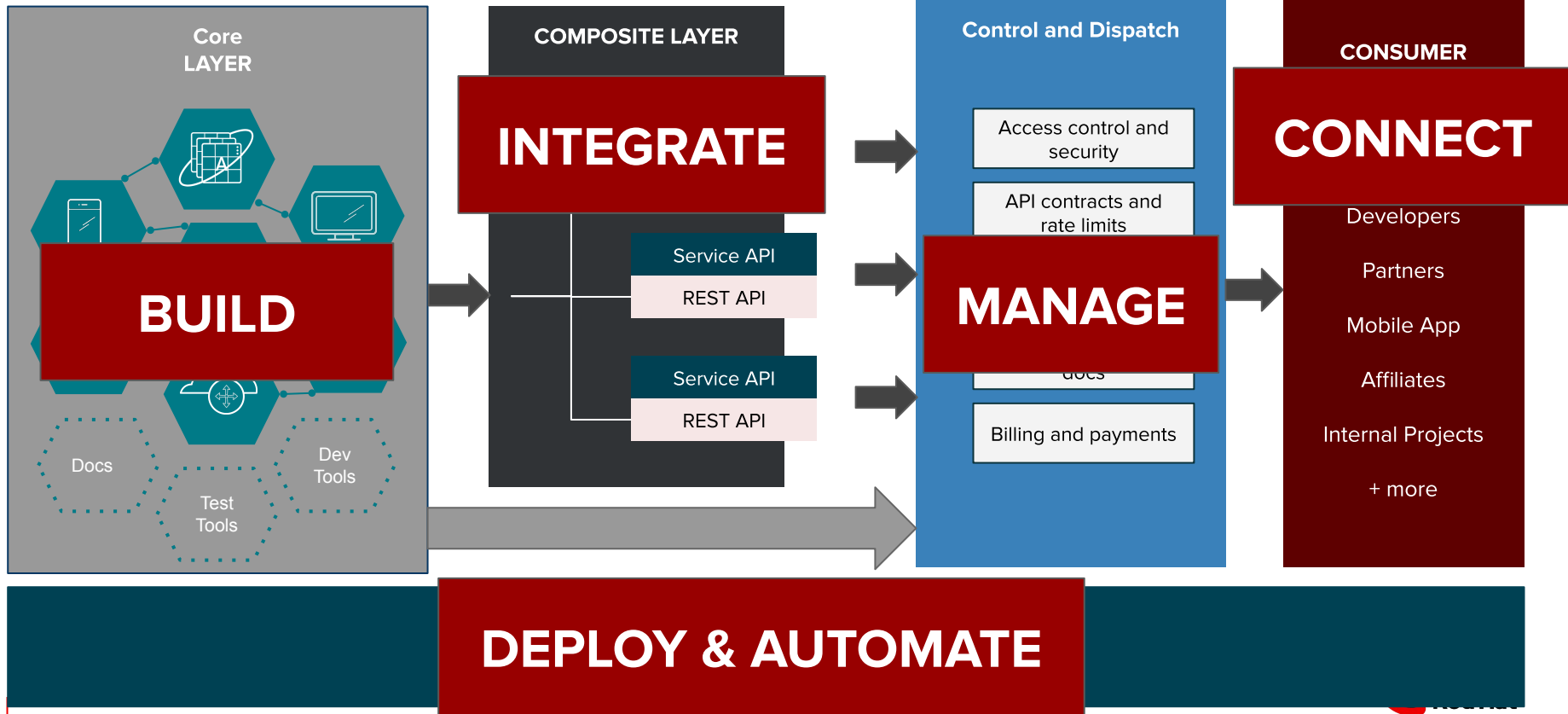
DEMO TIME!






AGILE INTEGRATION SOLUTION STACK



ENABLING YOUR ENTERPRISE STRATEGY



TODO LIST

-  Avoid accidental API architectures
-  Understand the role of integration in API-centric architectures
-  Get “hands on” with Red Hat Integration

Thank you

Questions?



[linkedin.com/company/red-hat](https://www.linkedin.com/company/red-hat)



[youtube.com/user/RedHatVideos](https://www.youtube.com/user/RedHatVideos)



[facebook.com/redhatinc](https://www.facebook.com/redhatinc)



twitter.com/RedHat