



# MuleSoft®



## Composable Enterprise Architecture

API-led Connectivity Approach

Niket Mohey

Niket.Mohey@mulesoft.com

Sr. Solutions Consultant





Frustrated by integration "donkey work", Ross Mason, VP Product Strategy, founded the open source Mule project in 2003. He created a new platform that emphasized ease of development with quick and efficient assembly of components, instead of custom-coding by hand.

The story behind the name





# THE CASE FOR A COMPOSABLE ENTERPRISE

# The world has changed



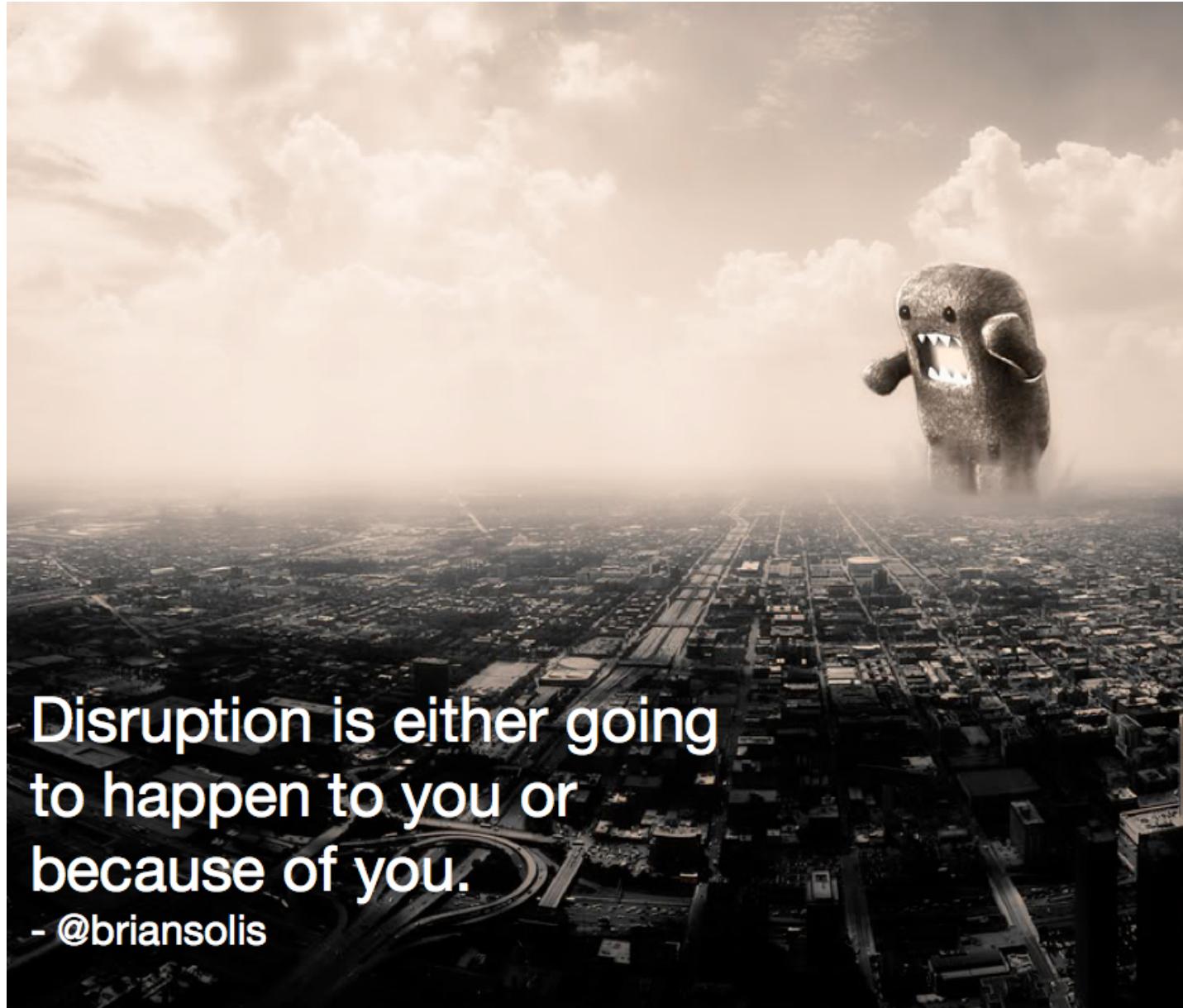
Only 12% of the 1955 Fortune 500 firms were operating in 2015.

The average lifespan of a company on the Fortune 500 has decreased from 90 years in 1935 to 18 years today.

57% of technology budgets are spent on maintaining legacy

86% of ITDMs say they are under “moderate” to “extreme” pressure to deliver IT services faster than they did last year.

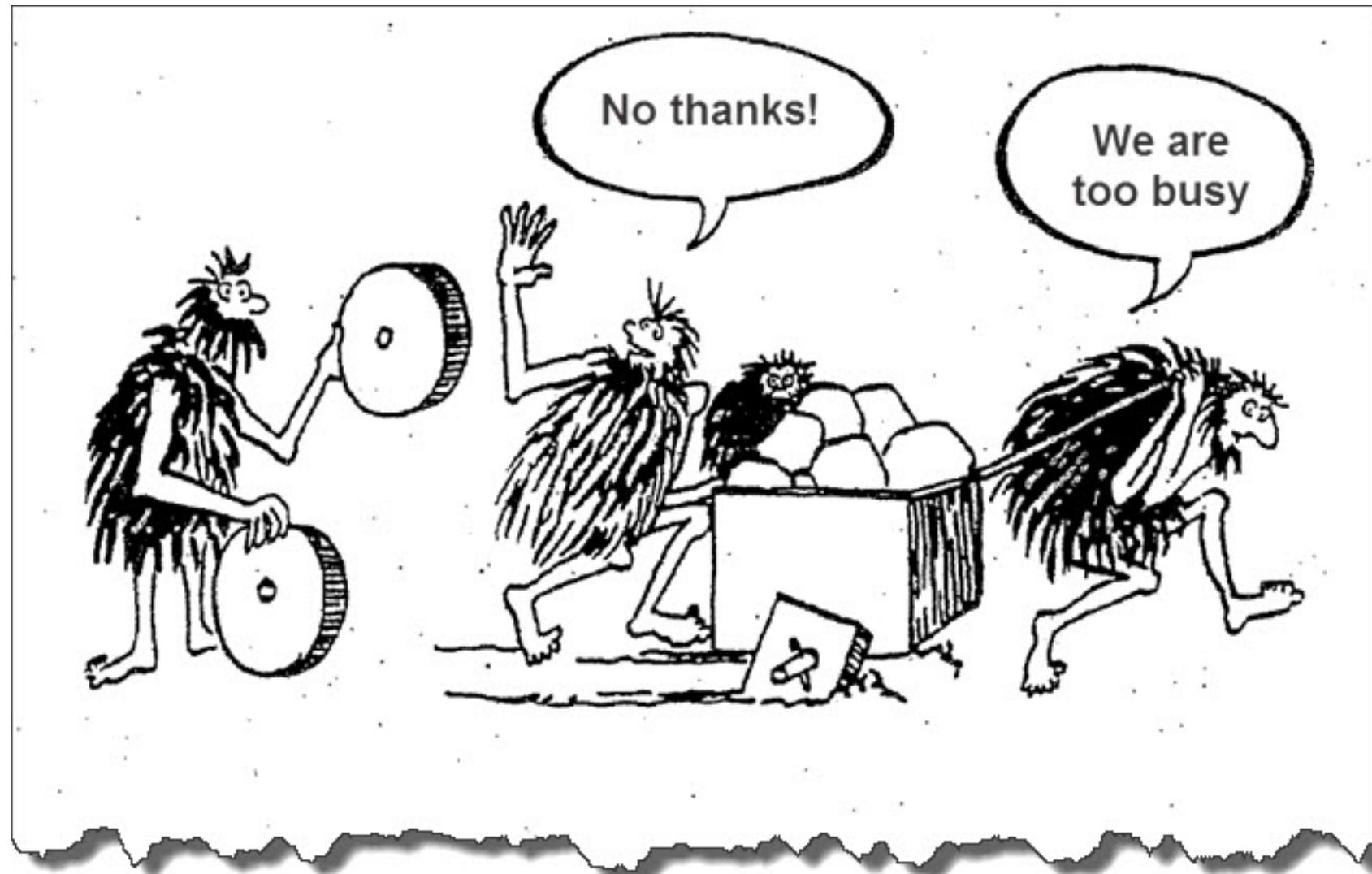
# The forecast is dire...



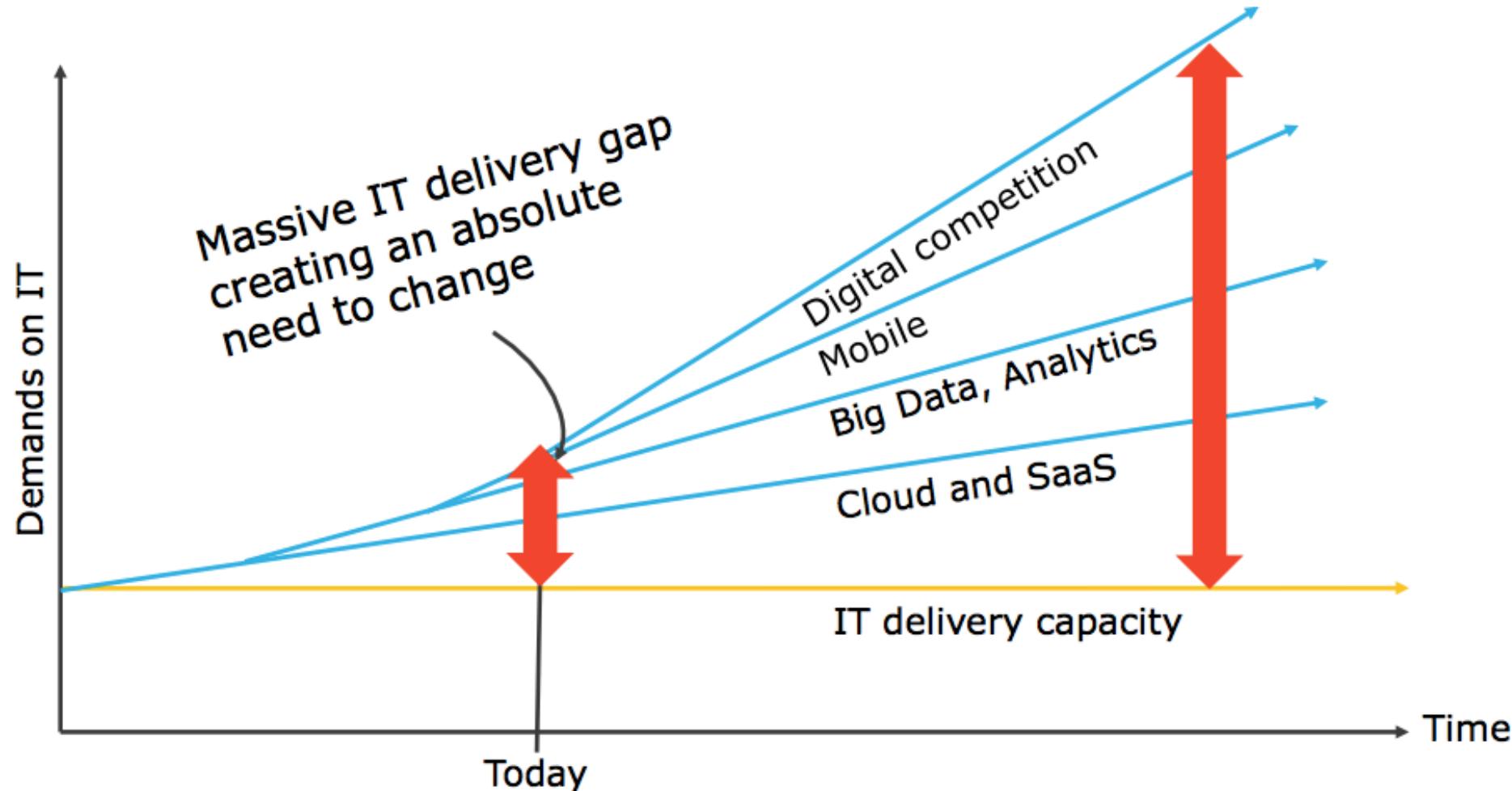
**Disruption is either going  
to happen to you or  
because of you.**

- @briansolis

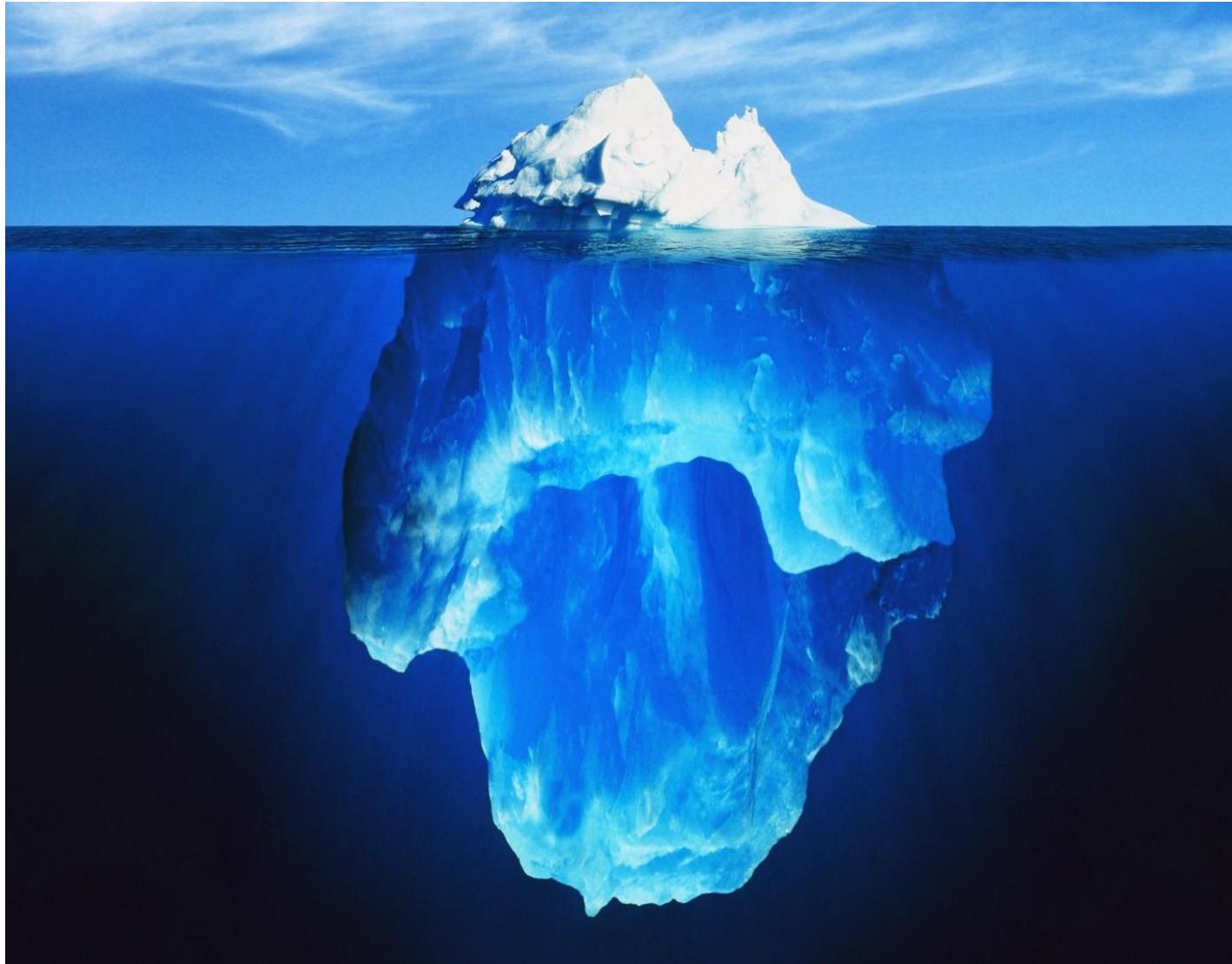
# The status quo won't suffice



# There is absolute imperative to change



# API Management vs. Connectivity



Connectors

Routing

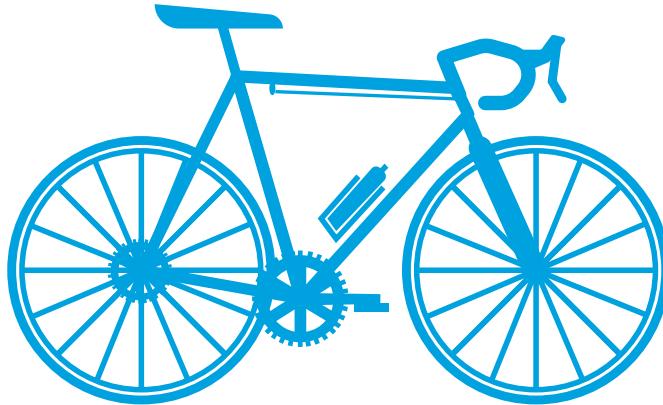
Transformation

---

Split/Aggregation

# Develop re-usable building blocks

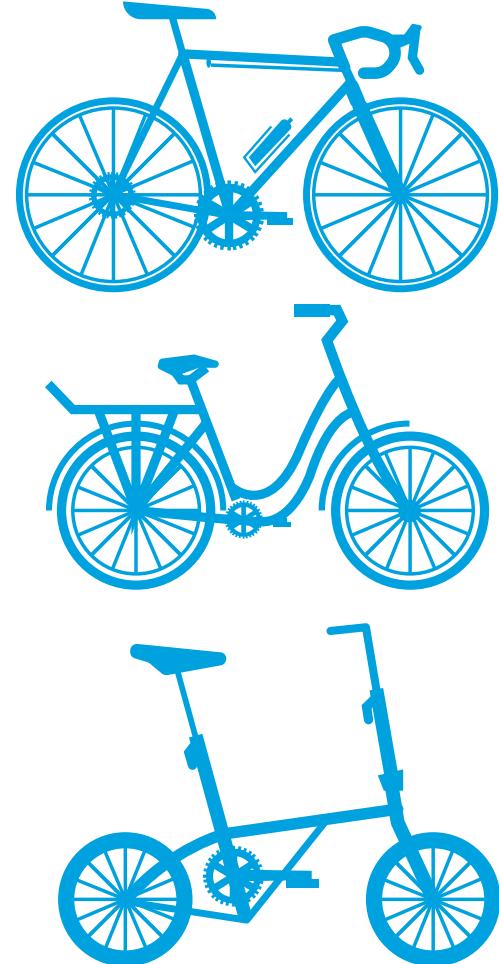
You have this



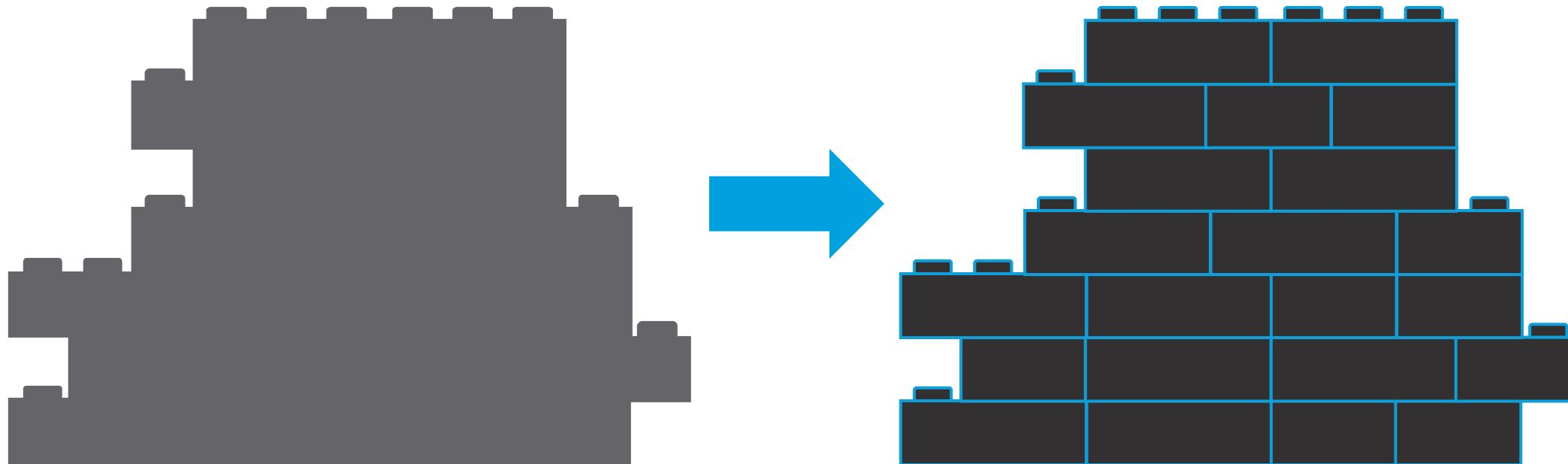
Your answer



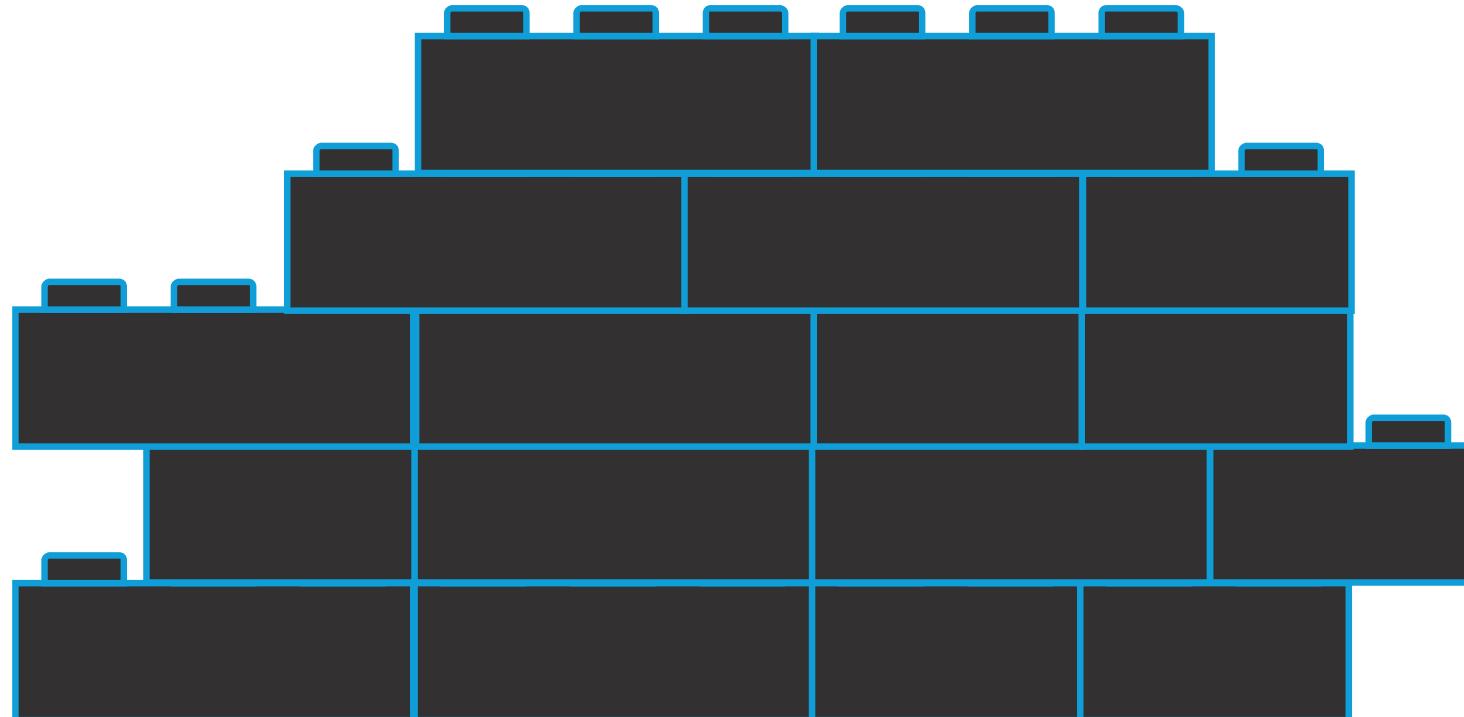
Business wants this



# Applications don't have to be monolithic



# Enable composability with APIs

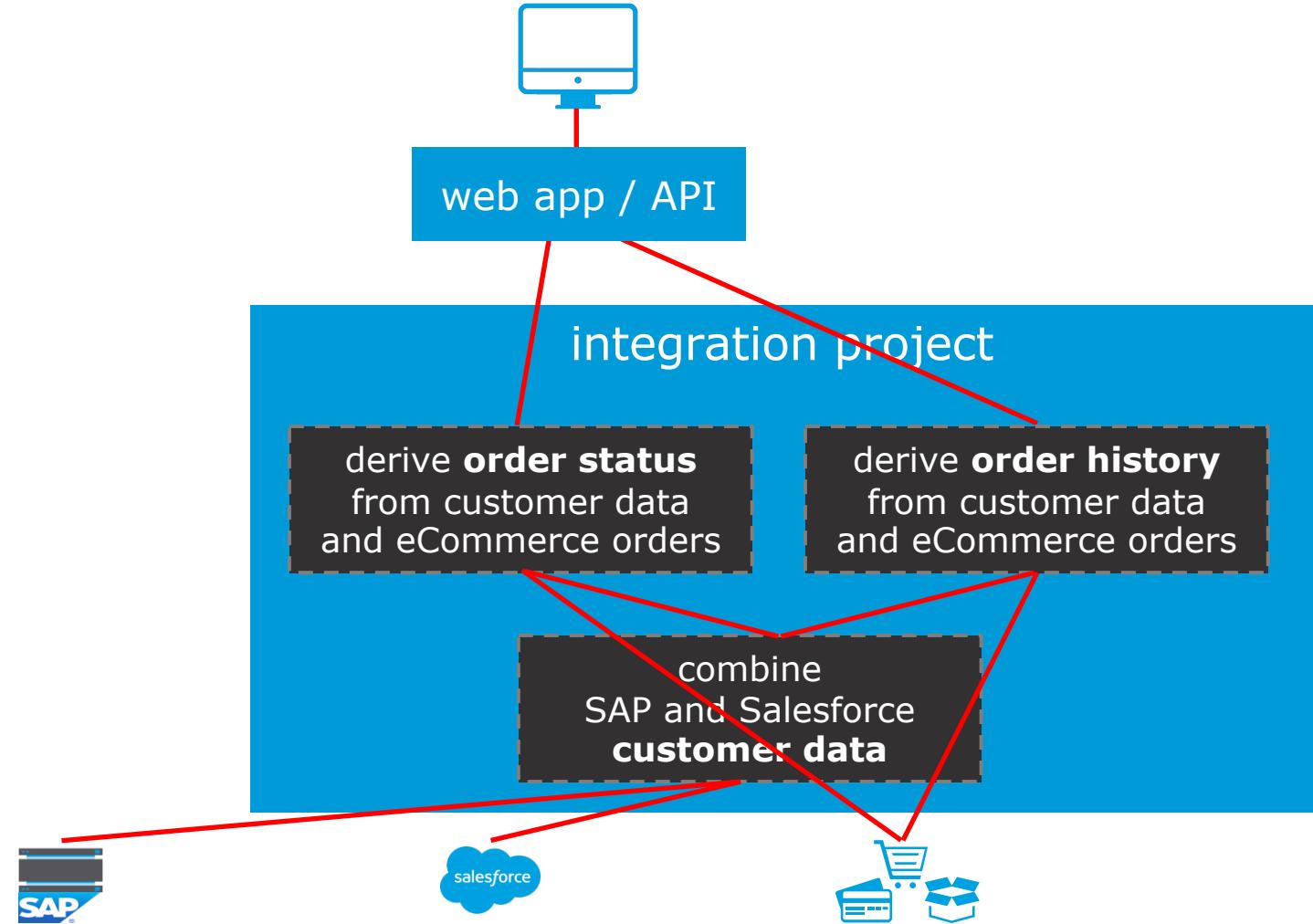


# A traditional project-based approach



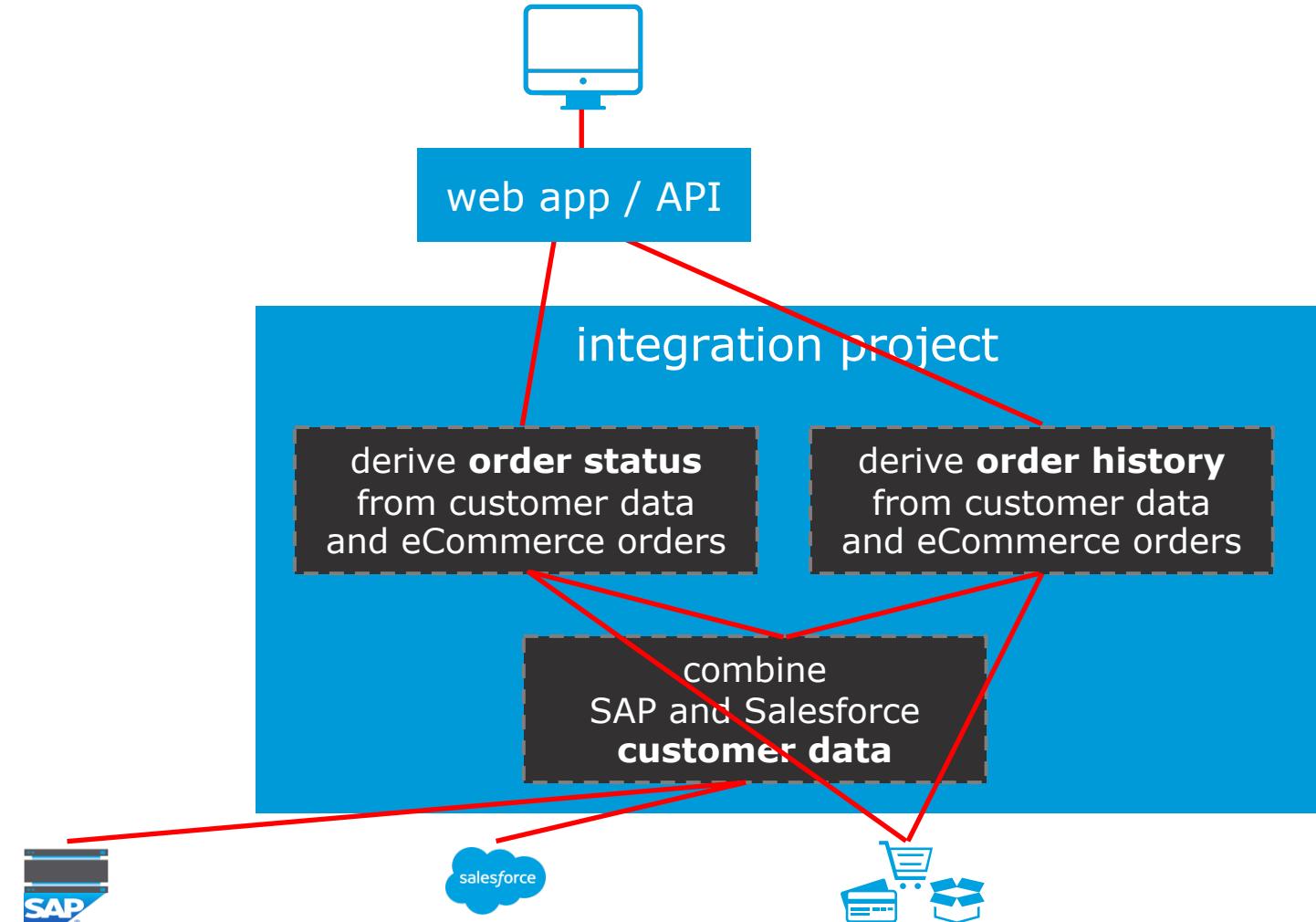
**Project objective:** Web app that provides real-time order status and order history for sales team engaging with customers

- Order data in eCommerce system
- Inventory data in SAP
- Customer data in SAP, SFDC

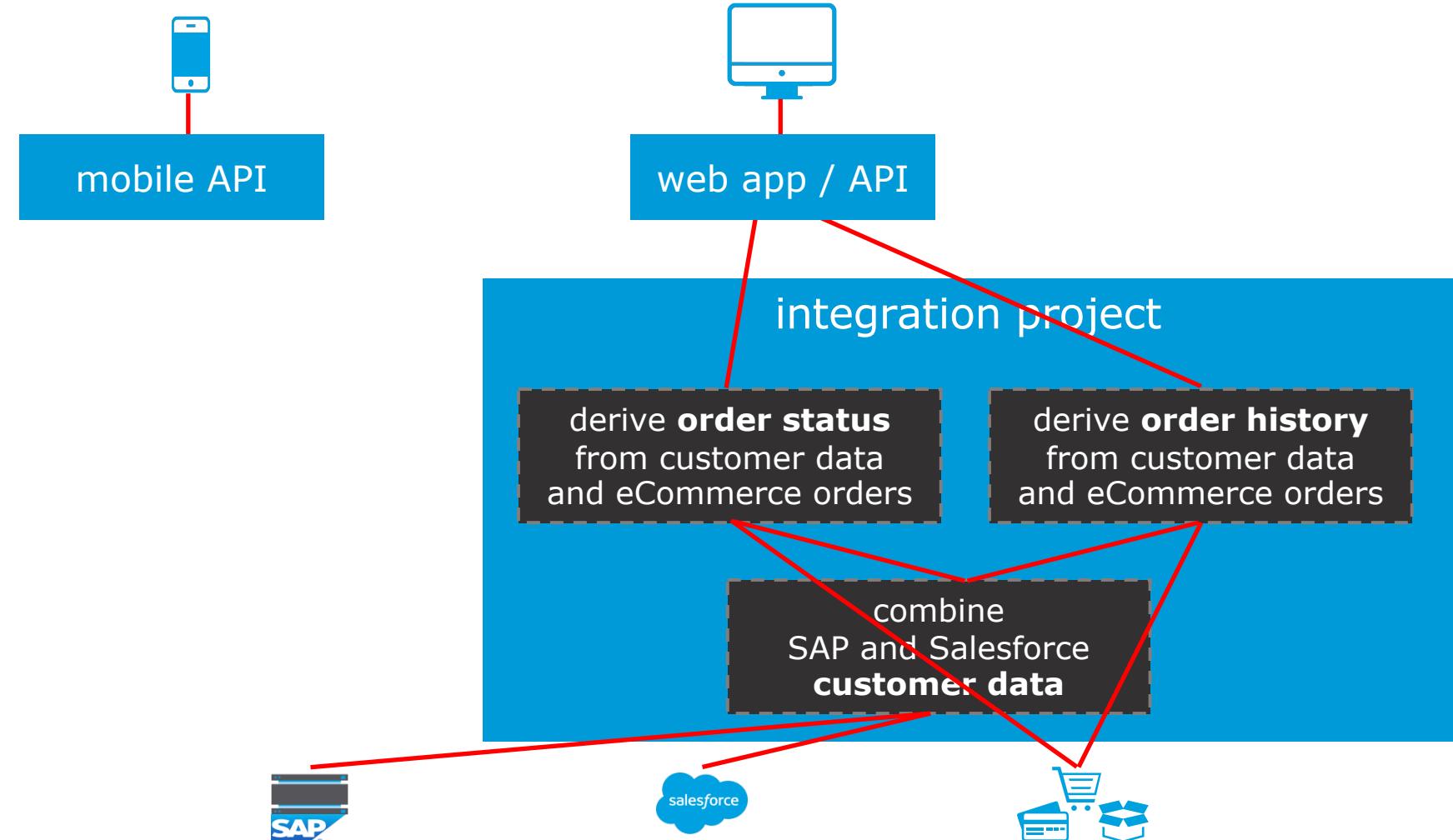


# A traditional project-based approach

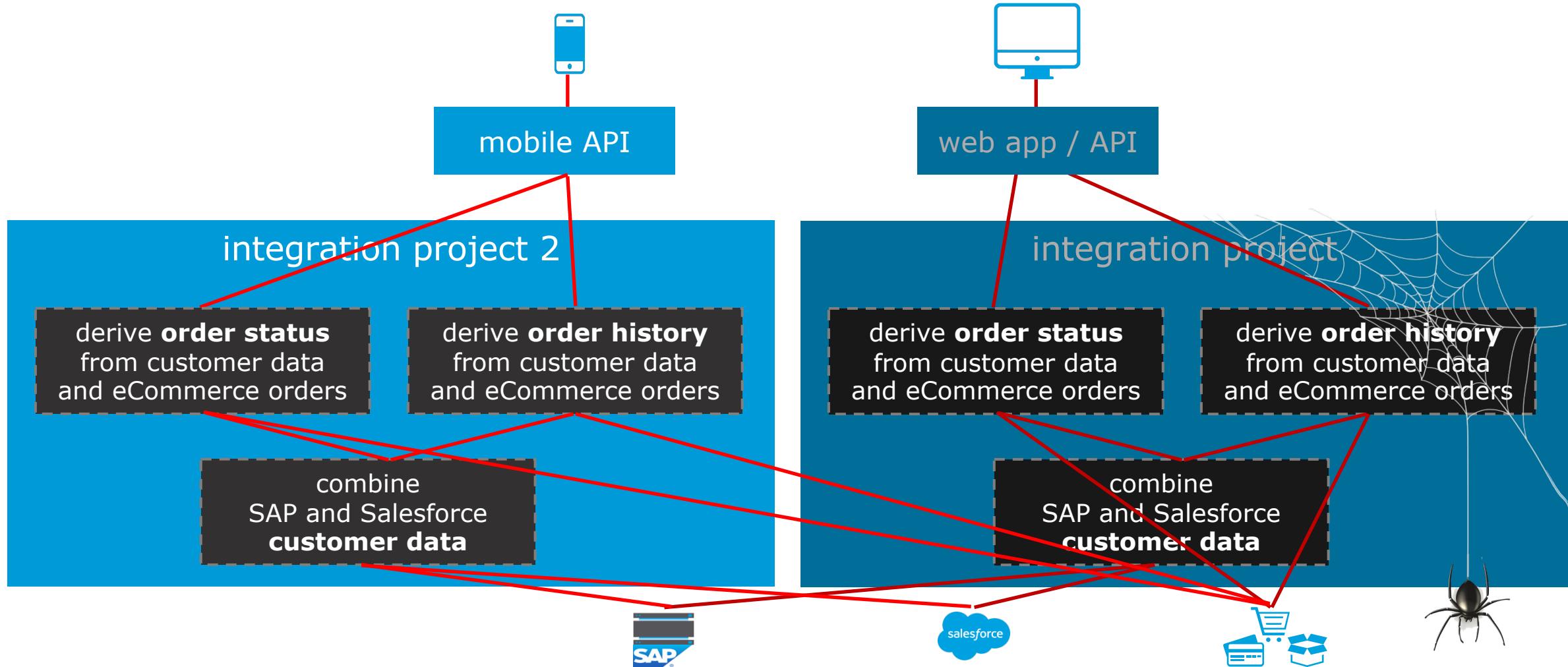
-  On time
-  Within budget
-  Meets requirements
-  Reuse
-  Loosely coupled to apps
-  Governance



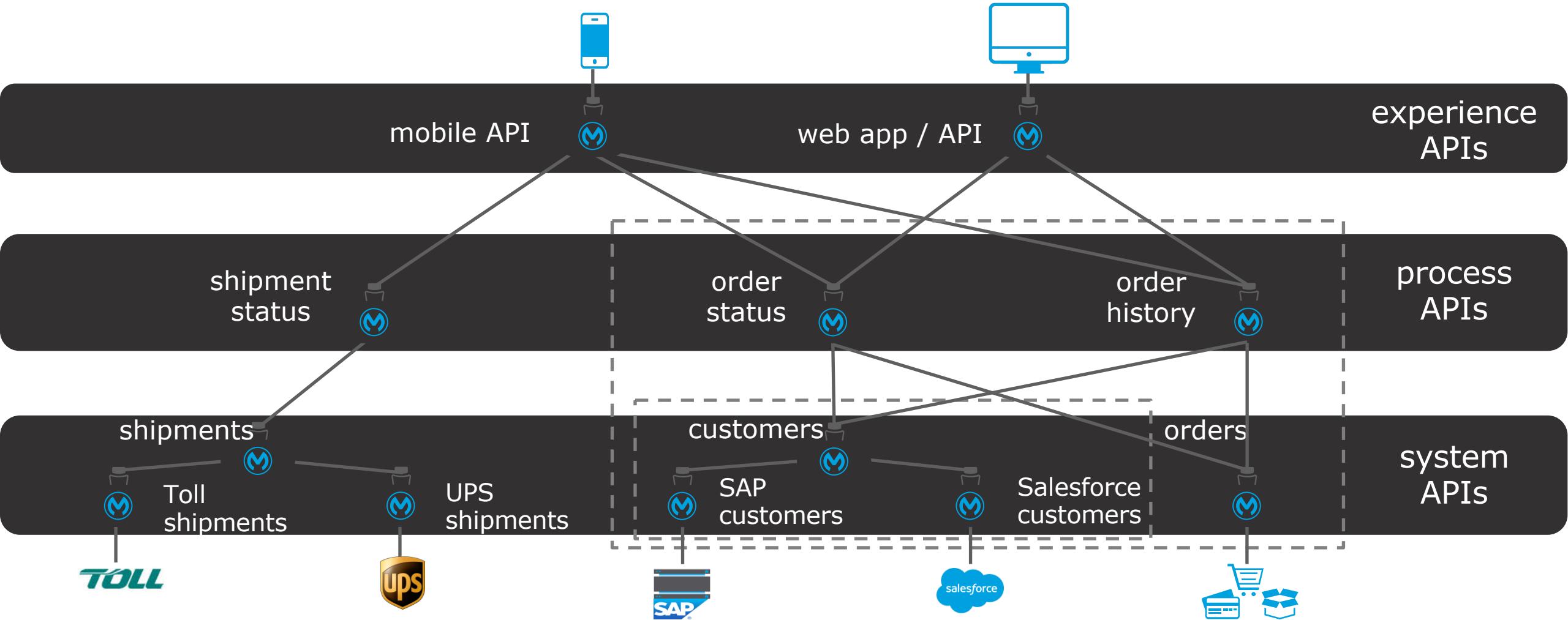
# 6 months later...



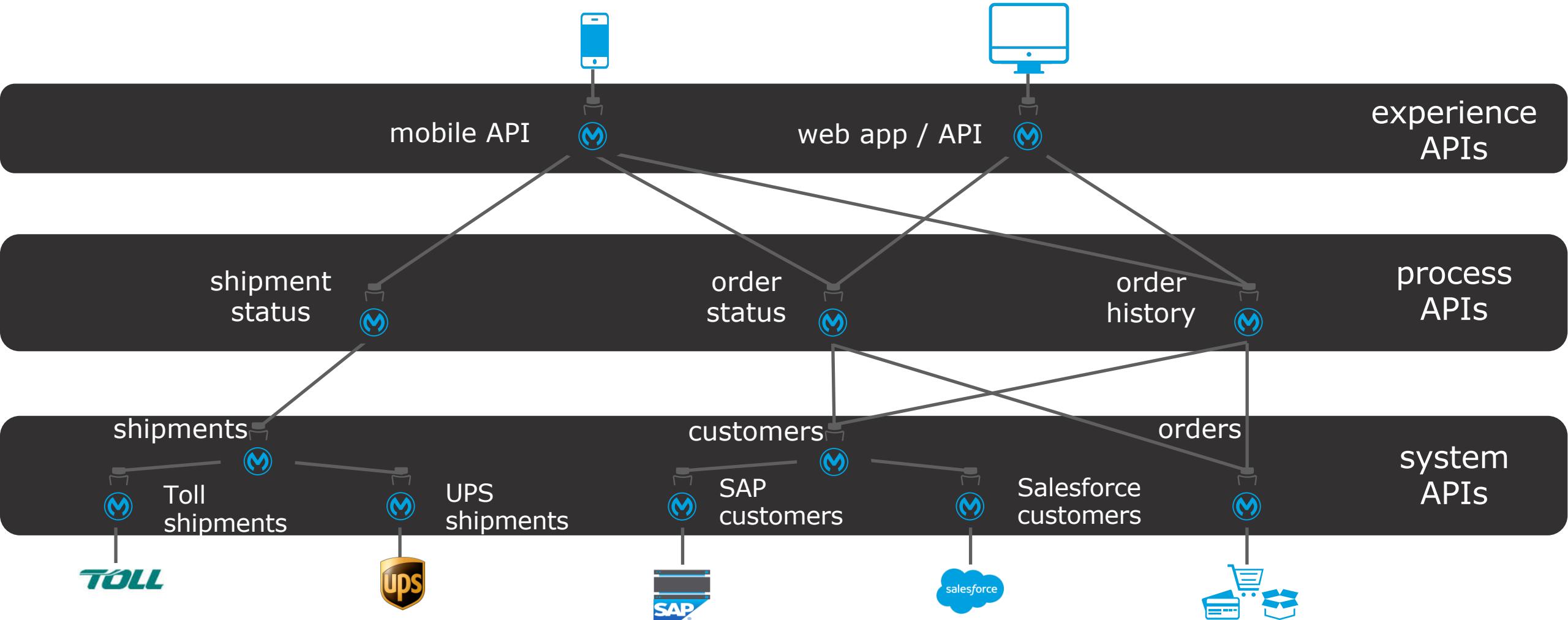
# 6 months later...



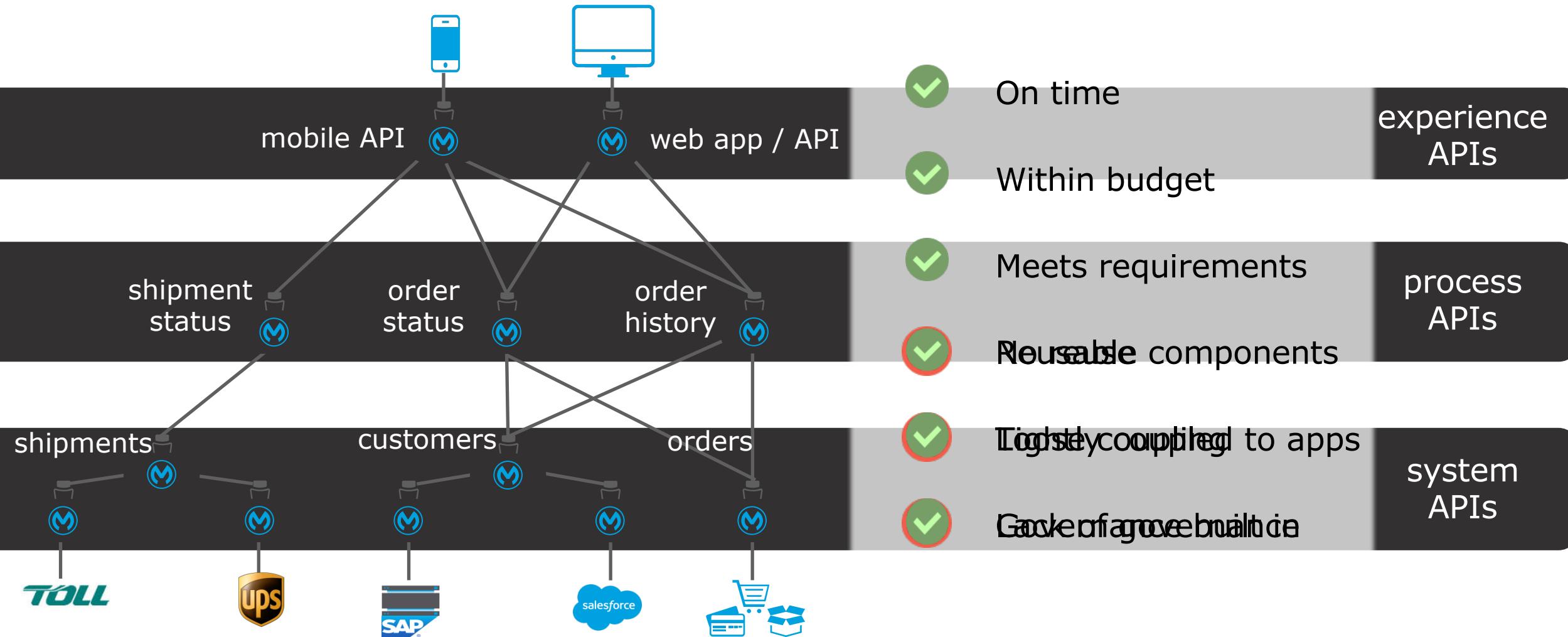
# The API-led connectivity approach



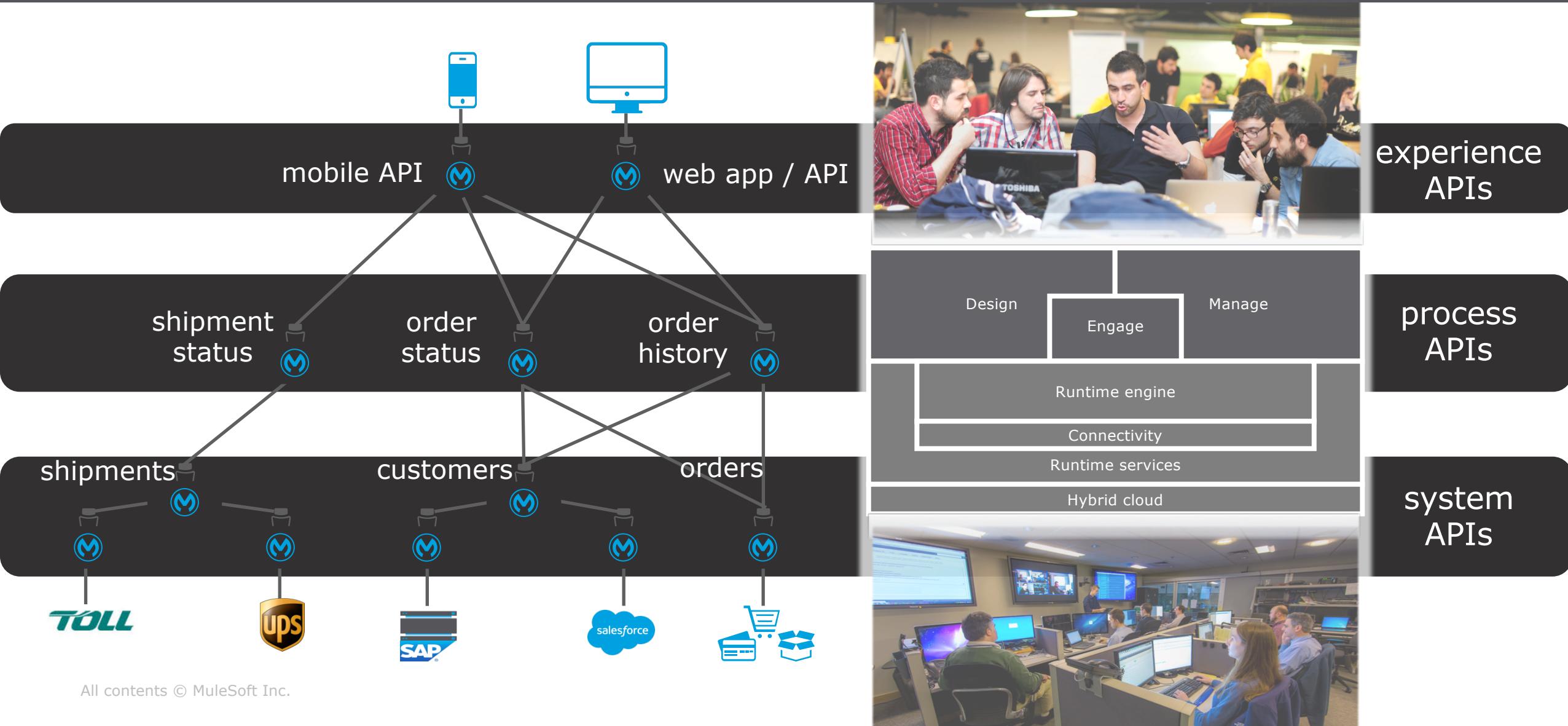
# The API-led connectivity approach



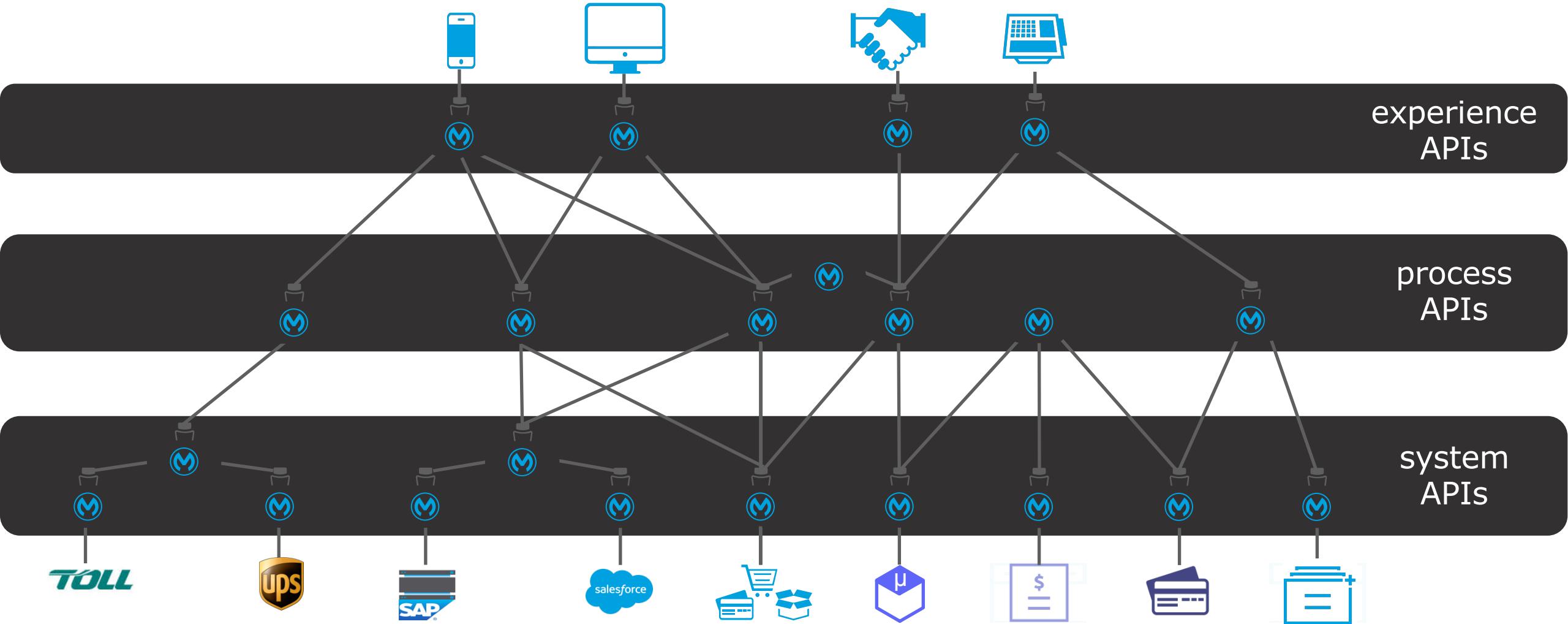
# The API-led connectivity approach



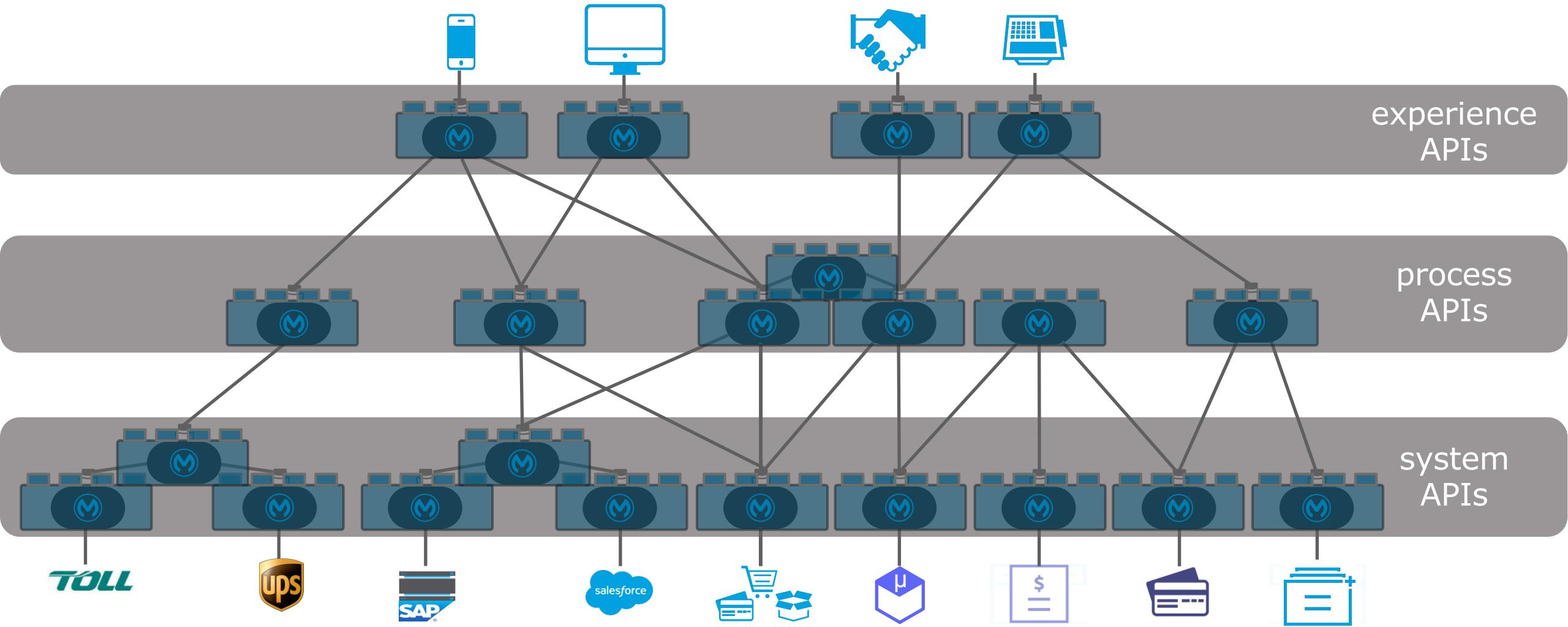
# The API-led connectivity approach



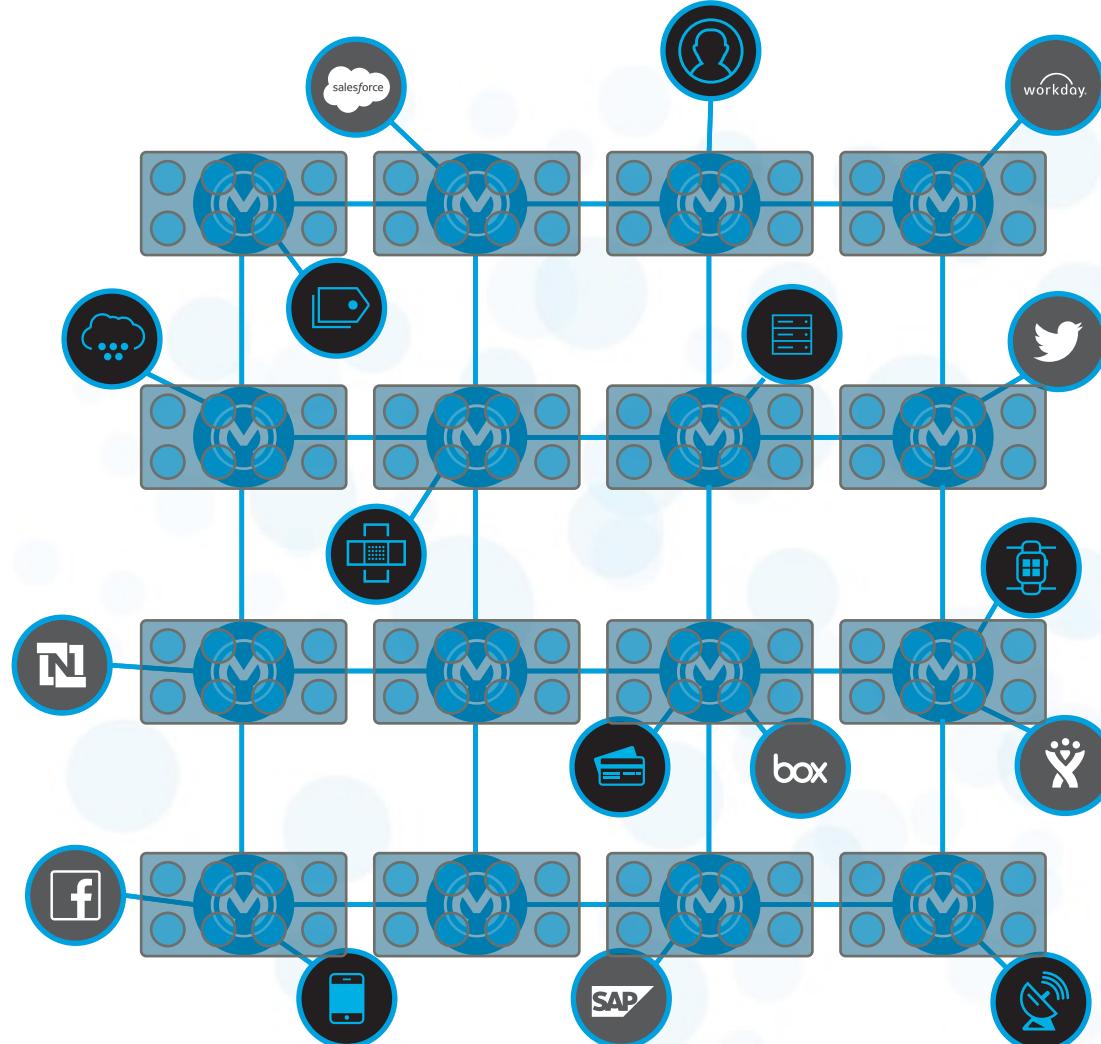
# API-led connectivity: future-proof



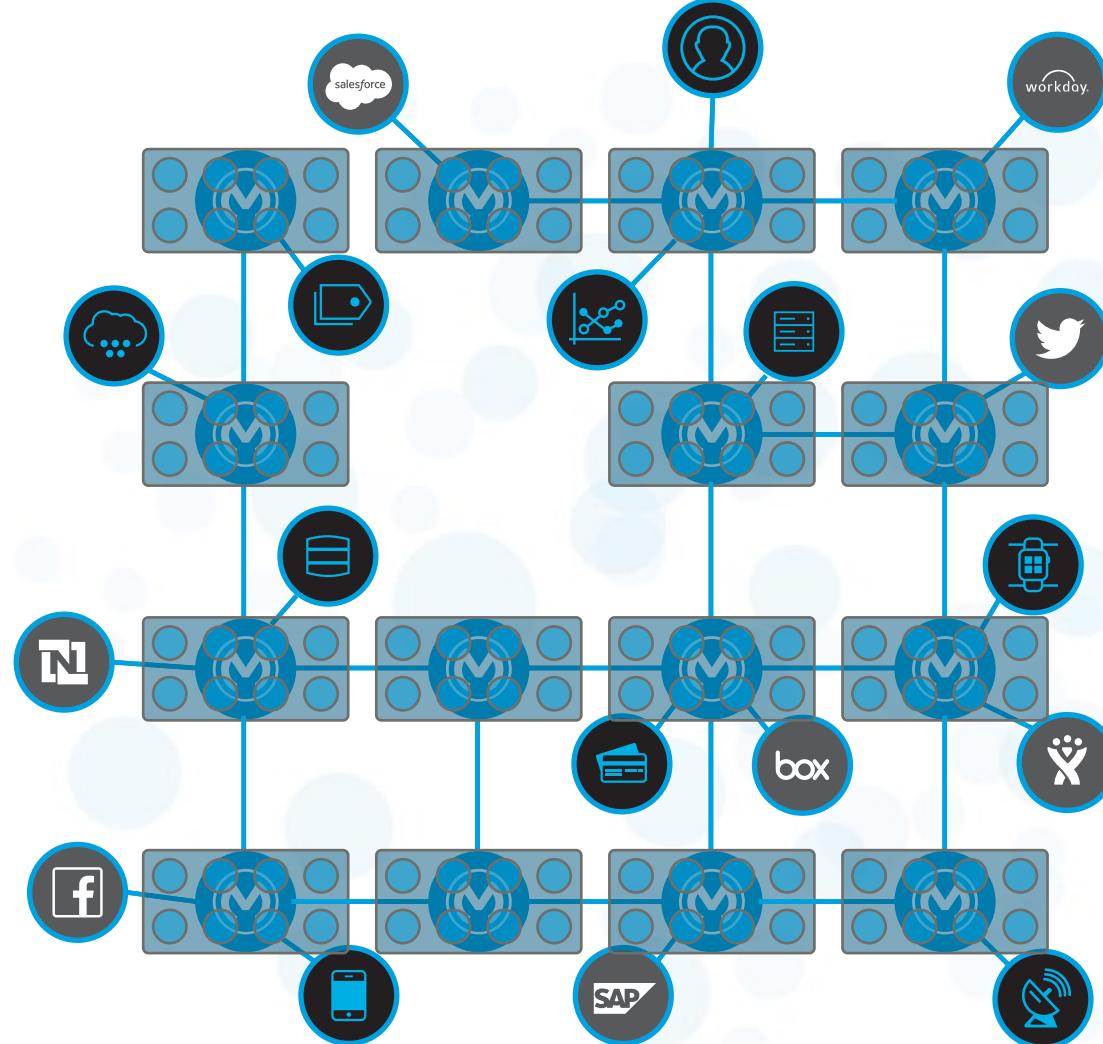
# API-led connectivity: building blocks



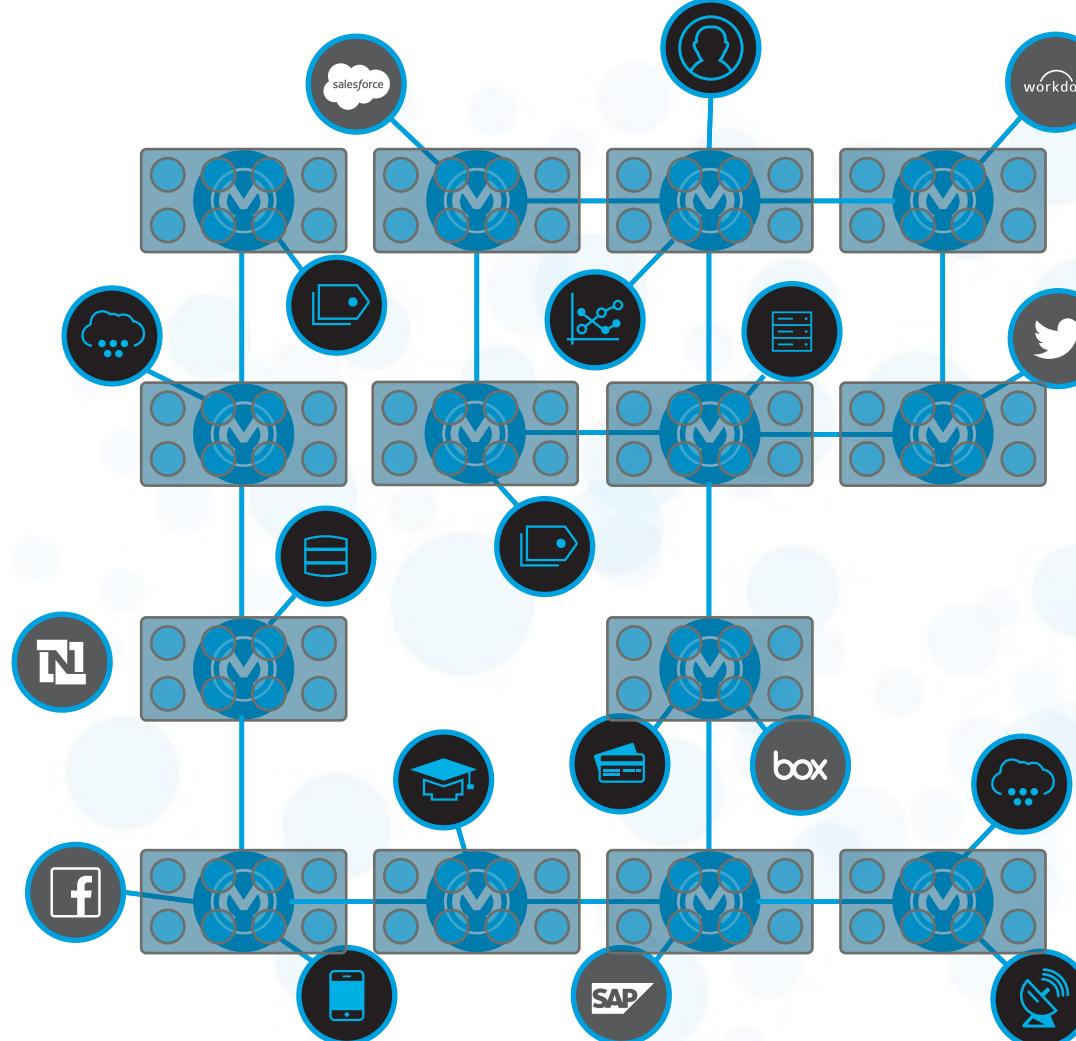
# The application network emerges



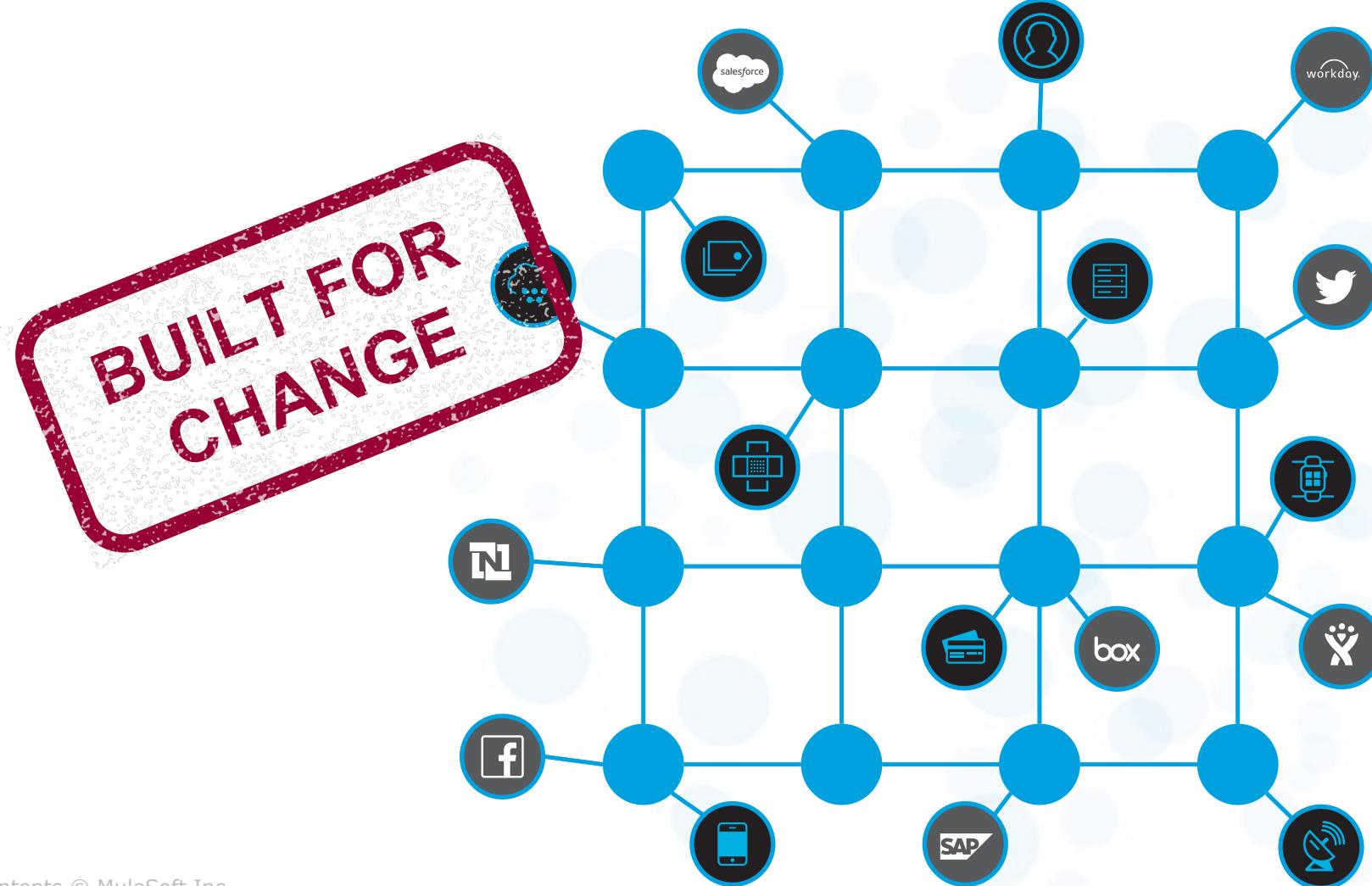
# Lather, rinse, repeat

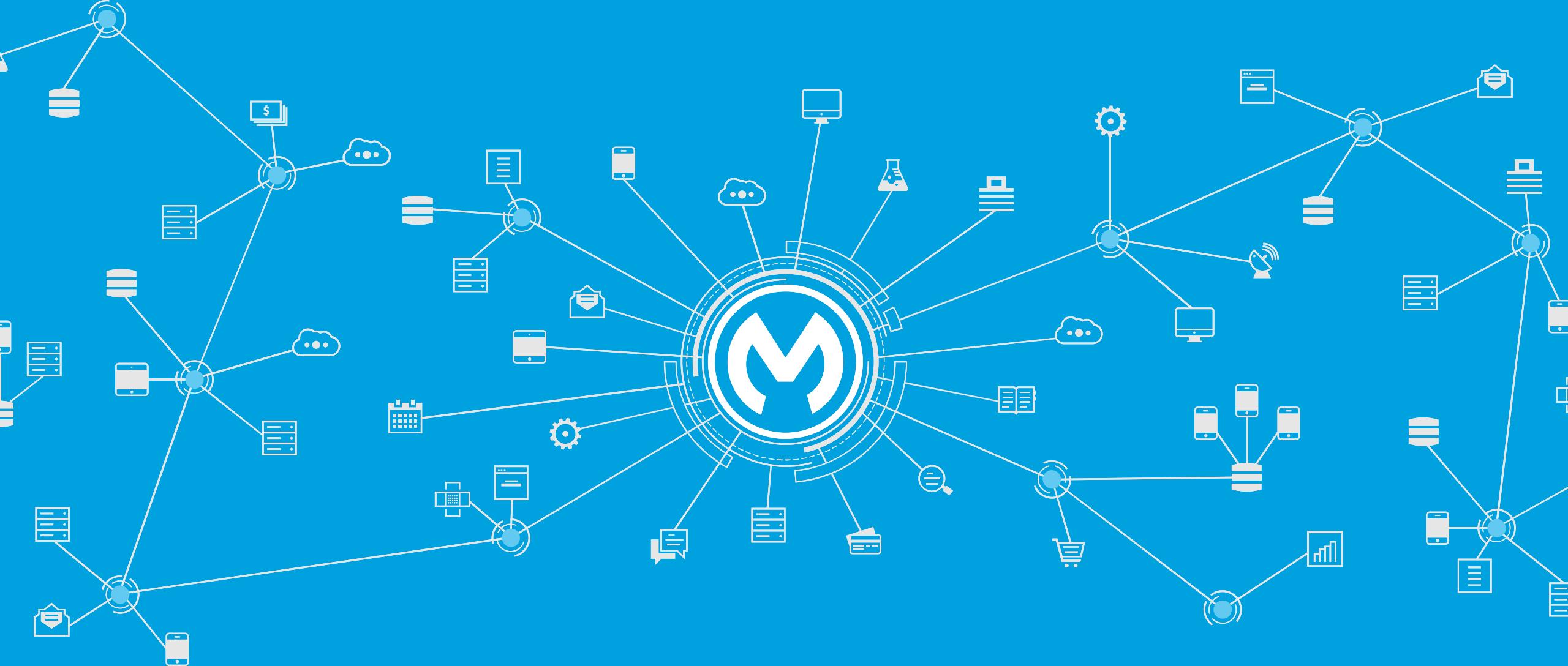


# Lather, rinse, repeat



# Your application network





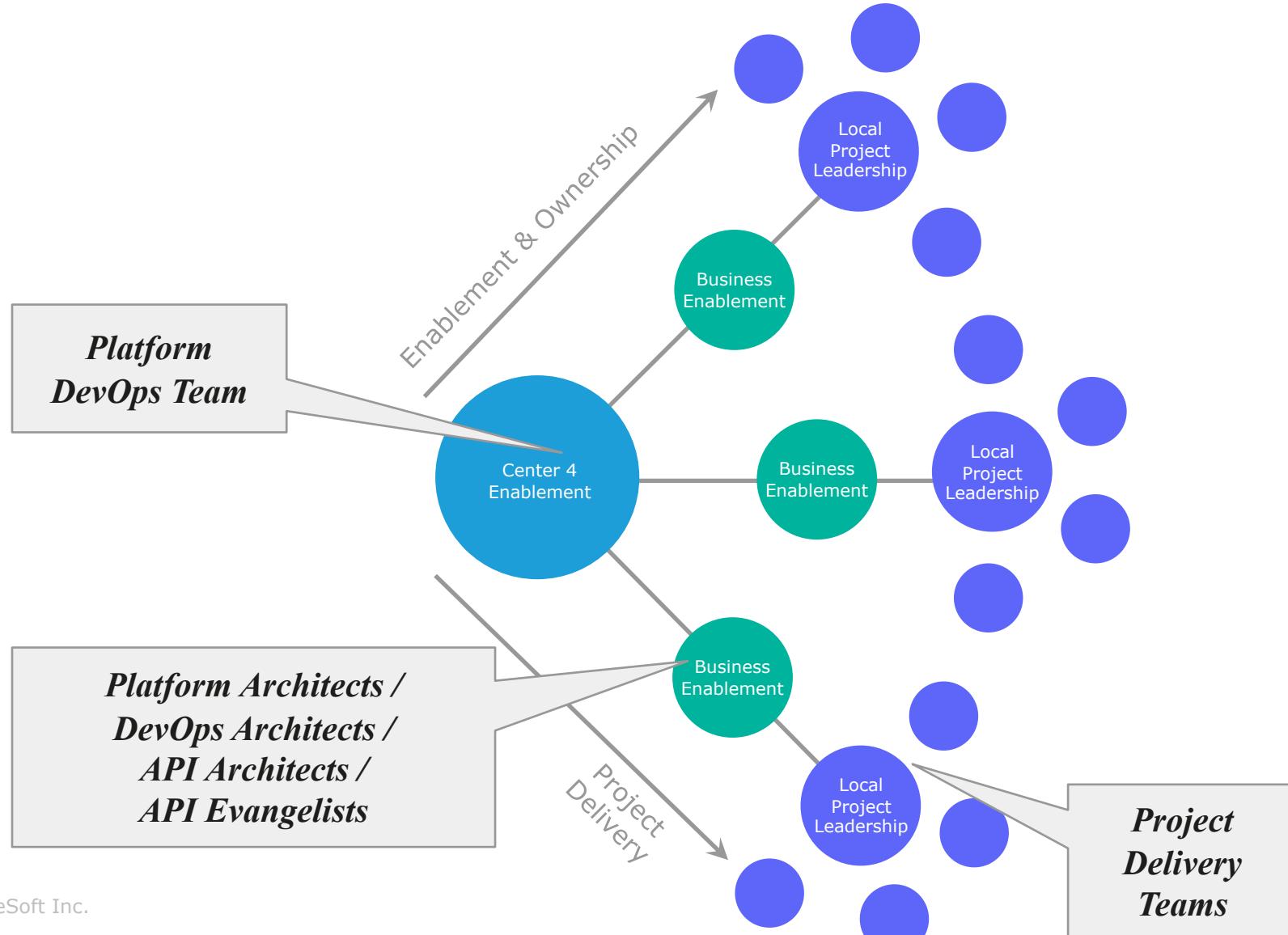
# How ??



## The C4E is a new operating model for IT

- Enablement over delivery
- Not an ivory tower!
- Decentralize to scale
- Drive multi-modal / multi-speed integration

# Achieving de-centralization, federation and speed Without losing control



- *Perceived* control over *all* technology assets
  - Implementing *everything*
  - *Order-taker* for the business
  - *Reacting* to business needs

- 
- *Federating* control to partners and lines of business
  - *Delegating* implementation to lines of business
    - *Enabling* self-service
    - *Proposing* business strategies

## Faster time to market

- Speeding up IT's response to threats and opportunities - by building capabilities that can be leveraged across the organization.
- Reducing delivery time for solutions, enabling Business or LoB IT teams to either build themselves, or configure without the need for coding.

## Build reusable IP with a set of core assets

- Providing reusable assets that can be consumed by audiences outside of IT, to allow businesses to self-serve technology projects.
- Retaining security and governance because these templates and APIs have security and governance built in.

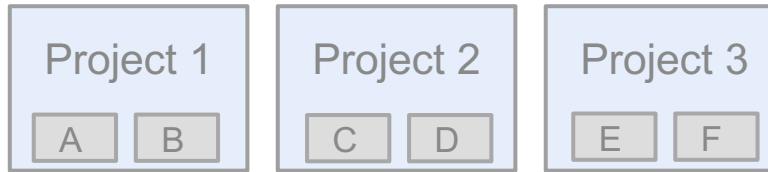
## Scaling across partners

- Reducing the pressure on IT to deliver every technology project, freeing up time to work on more strategic projects.
- Do things in a way that's consistent and common across your Partners, not dictated by them

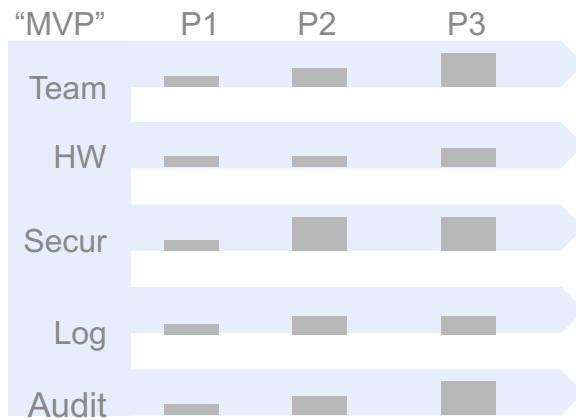
# The solution: a balance of project and platform focus



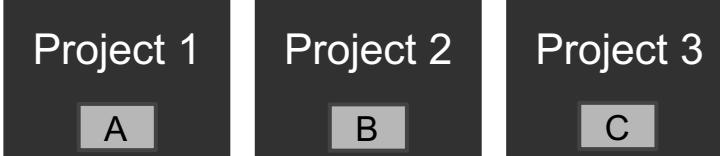
## Option 1: Project-led



Project Timeline →

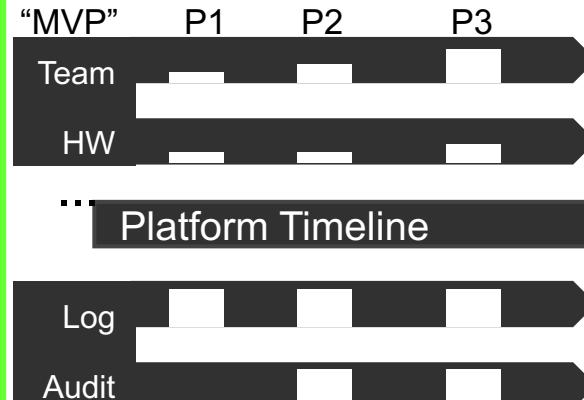


## Balanced Deployment

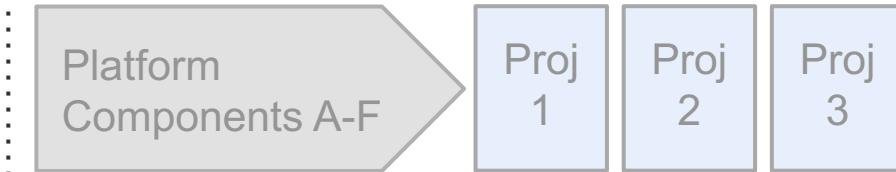


Use discovery process to identify foundational component sequencing

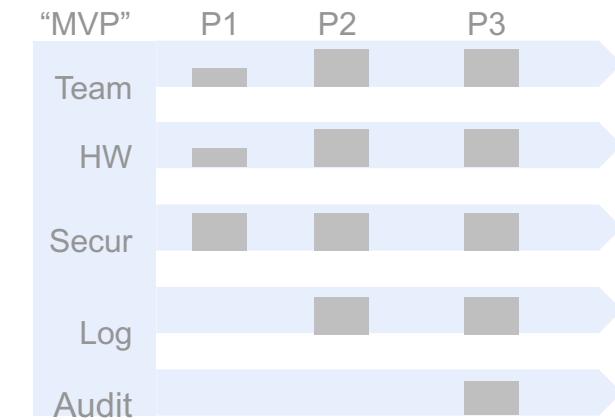
Project Timeline →

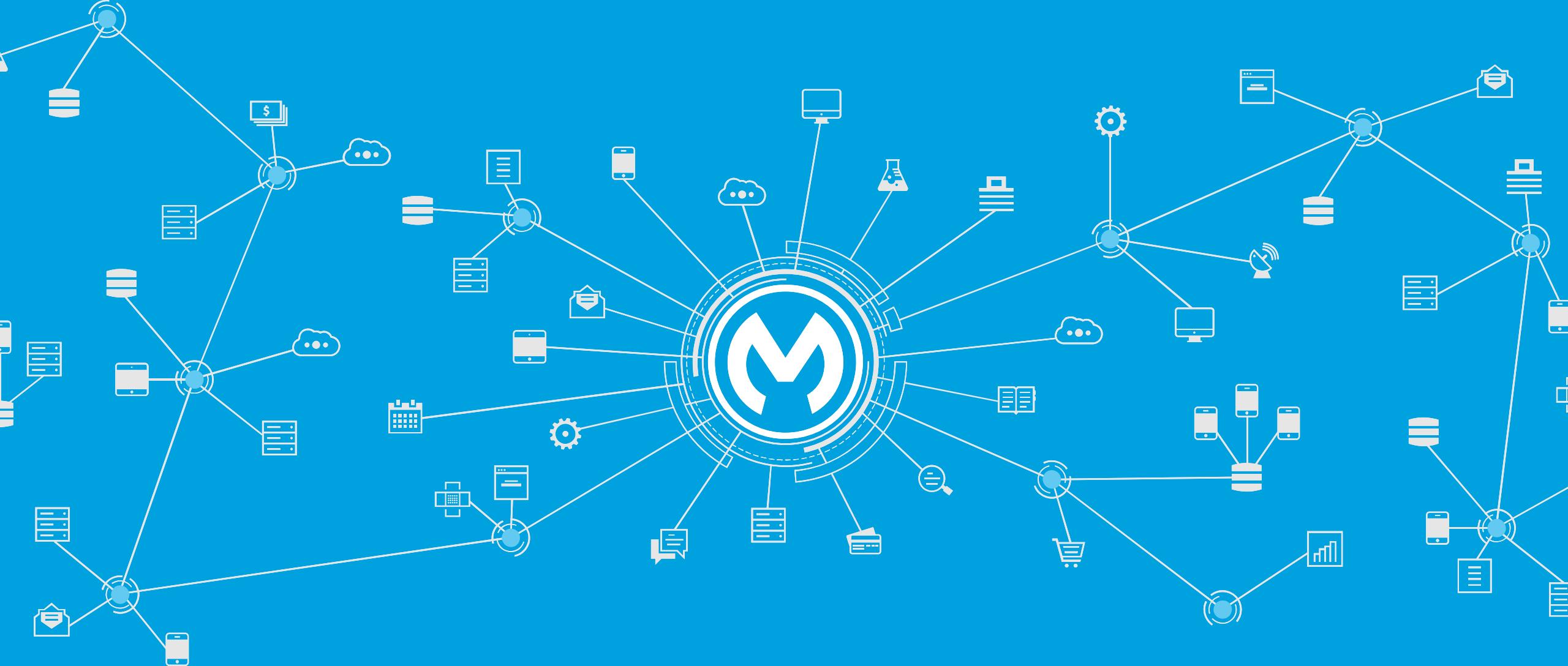


## Option 3: Platform-led



Platform Timeline → P1

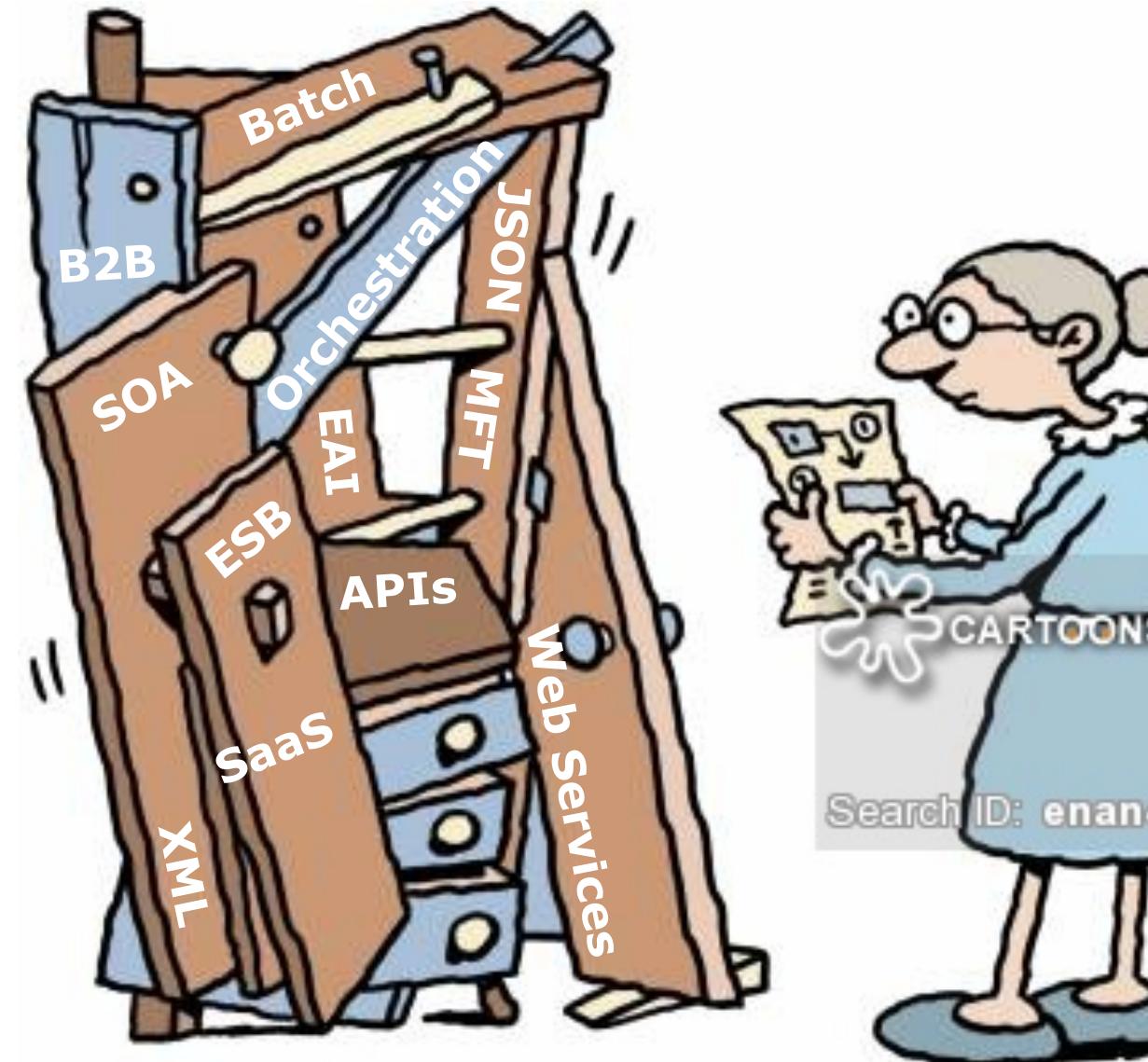




# Additional

# Why does Unified Connectivity matter?

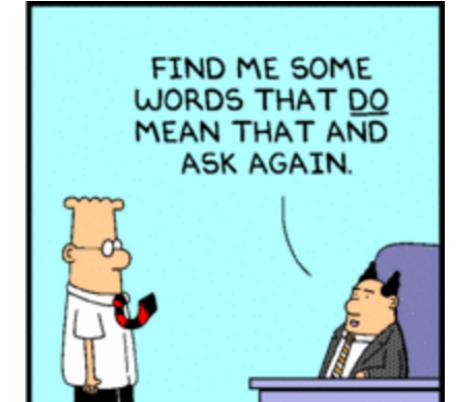
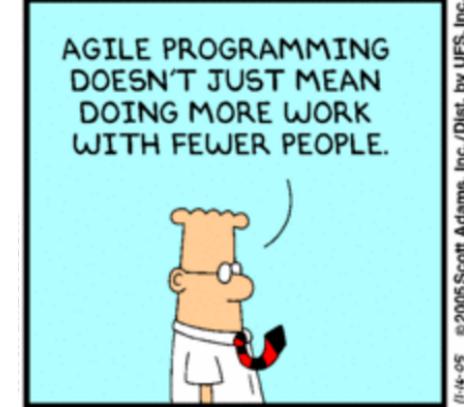
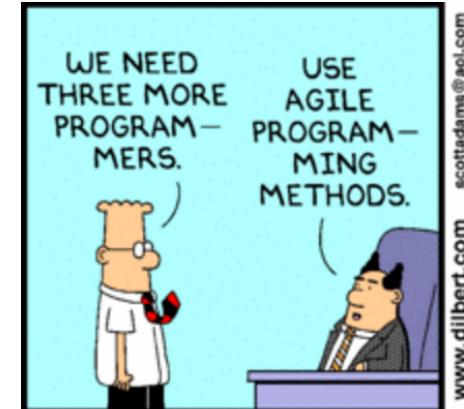
- The Hard Way
  - Different products for integration, SOA, APIs
  - Different development tools (usually proprietary)
  - Different management tools for each piece
- Anypoint Platform
  - **Single runtime**, run on cloud, on-prem, on your laptop
  - **Single development** environment that supports the common open source dev tools
  - **Single management** environment across SOA, SaaS and API connectivity



# What makes Anypoint Platform more productive?



- The Hard Way
  - Proprietary tools that were built for a different era
  - Have their own SDLC
  - Little or no support for automation or testing
- Anypoint Platform
  - Uses/integrations **common Agile tools** that people are already using ie. Git, Maven, Jenkins
  - **Designed for productivity** i.e. every step from download, build, deploy, change is designed to be fast
  - Built for **composability** – reusable building blocks, connectors and templates
  - Run local instantly, run in the cloud in seconds
  - Developers & consumers of APIs **collaborate** with RAML and build for purpose



- The Hard Way
  - Different technologies for cloud and on-premises
  - Black box: difficult or impossible to extend
  - Fixed architecture
- Anypoint Platform
  - **Flexible architecture** models that can be molded to your needs
  - **Hybrid**, Build once, run anywhere - deploy applications to the Cloud or on-premises
  - Enterprise grade connectivity and global cloud means we **scale** for every size enterprise
  - **Extensible** – support new applications/technologies through DevKit

