COPENHAGEN
INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE 2014

goto; conference

SHOOTOUT

SERVICE VIRTUALIZATION

Roland Møller

CA Technologies

Me

Poor parents and two older sisters with bad taste

Cold war? Like in cold beer?

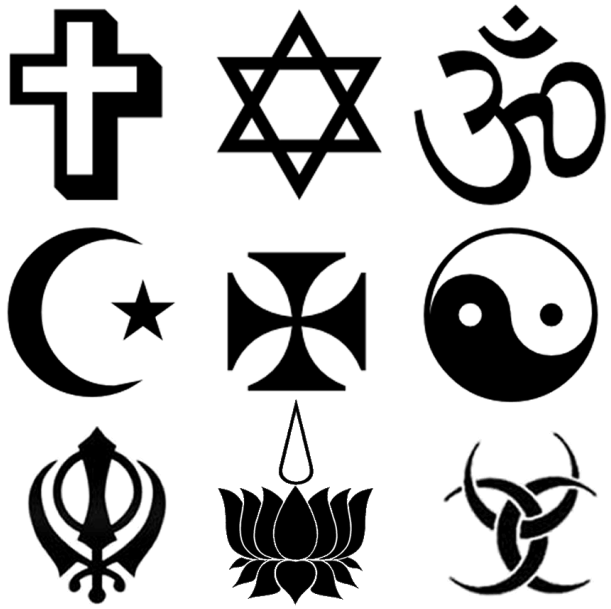
Service Virtualization (yeah right)

Definitions

Stubs provide canned answers to calls made during the test, usually not responding at all to anything outside what's programmed in for the test. Stubs may also record information about calls.

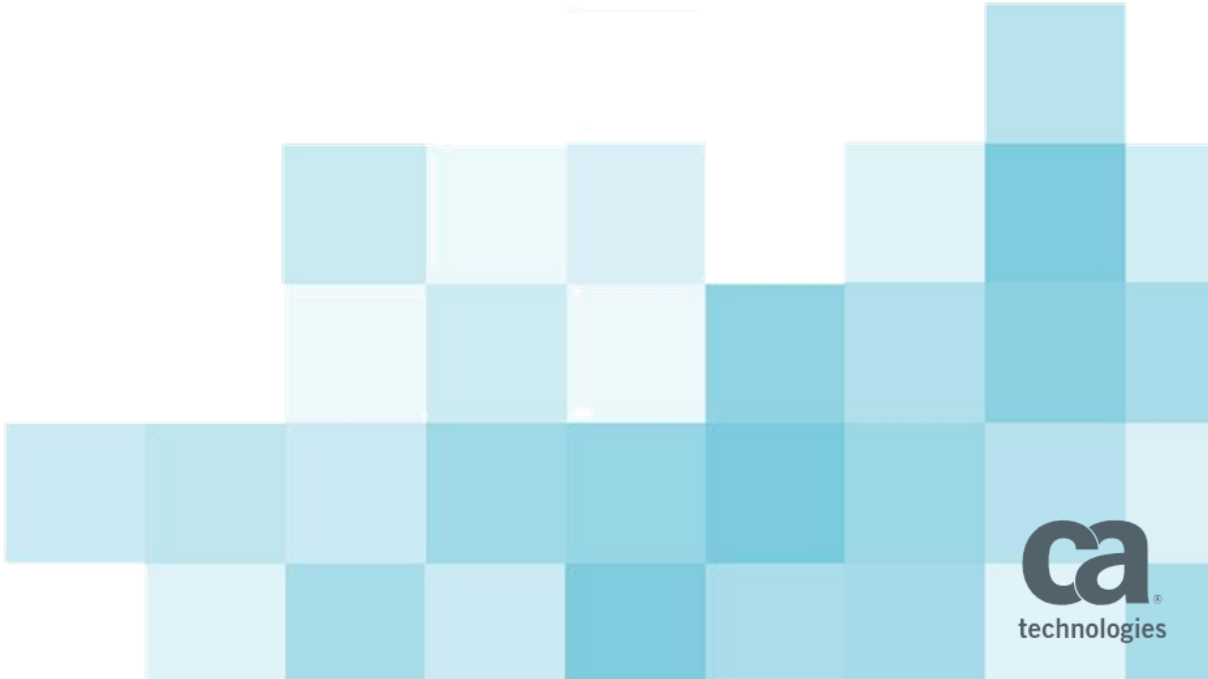
Mocks are objects pre-programmed with expectations which form a specification of the calls they are expected to receive.

Service Virtualization is a method of capturing and simulating the behavior of unavailable or incomplete systems.



What is Service Virtualization?

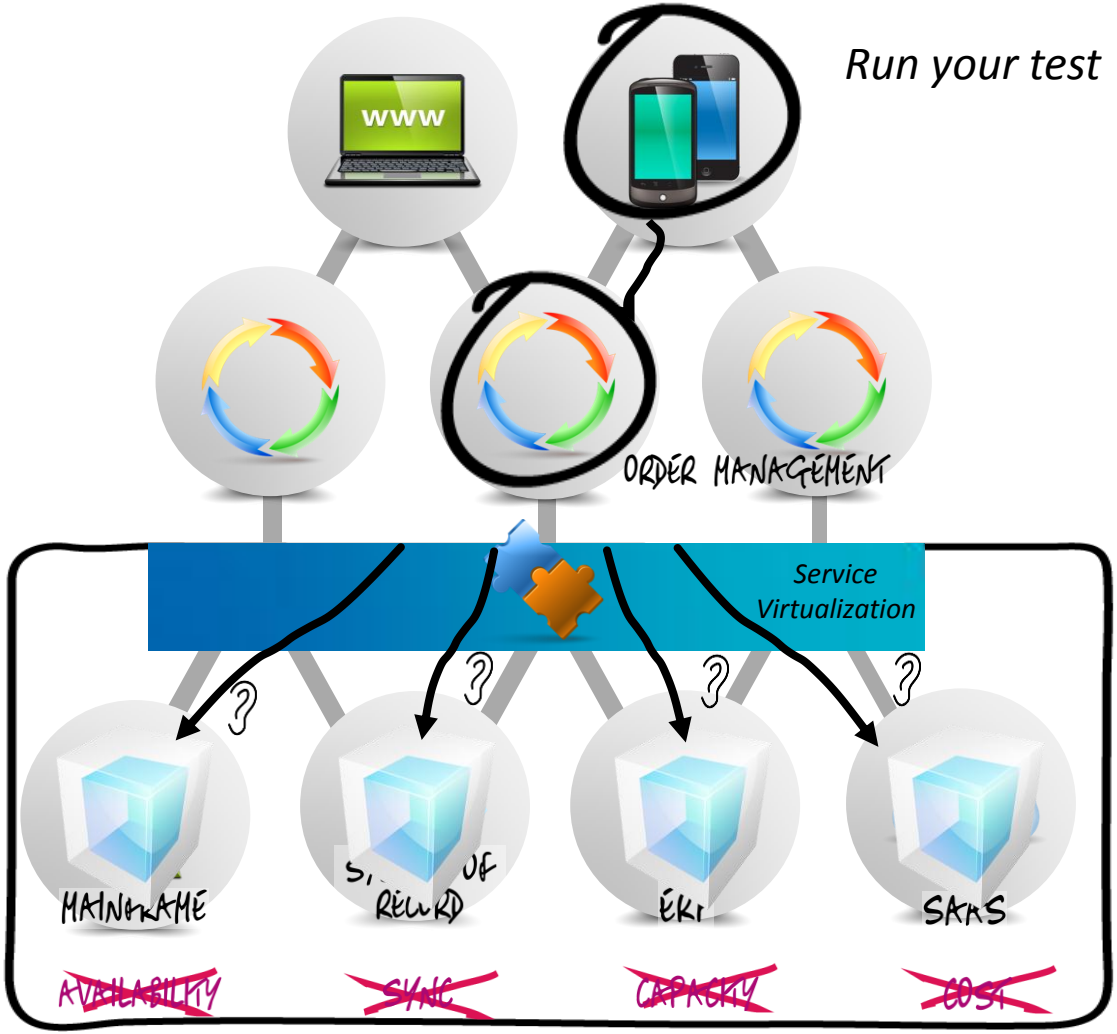
Stick man



Constraint Free Agile Development

SOLUTION: Service Virtualization

CONSTRAINT FREE DEVELOPMENT



PRESENTATION

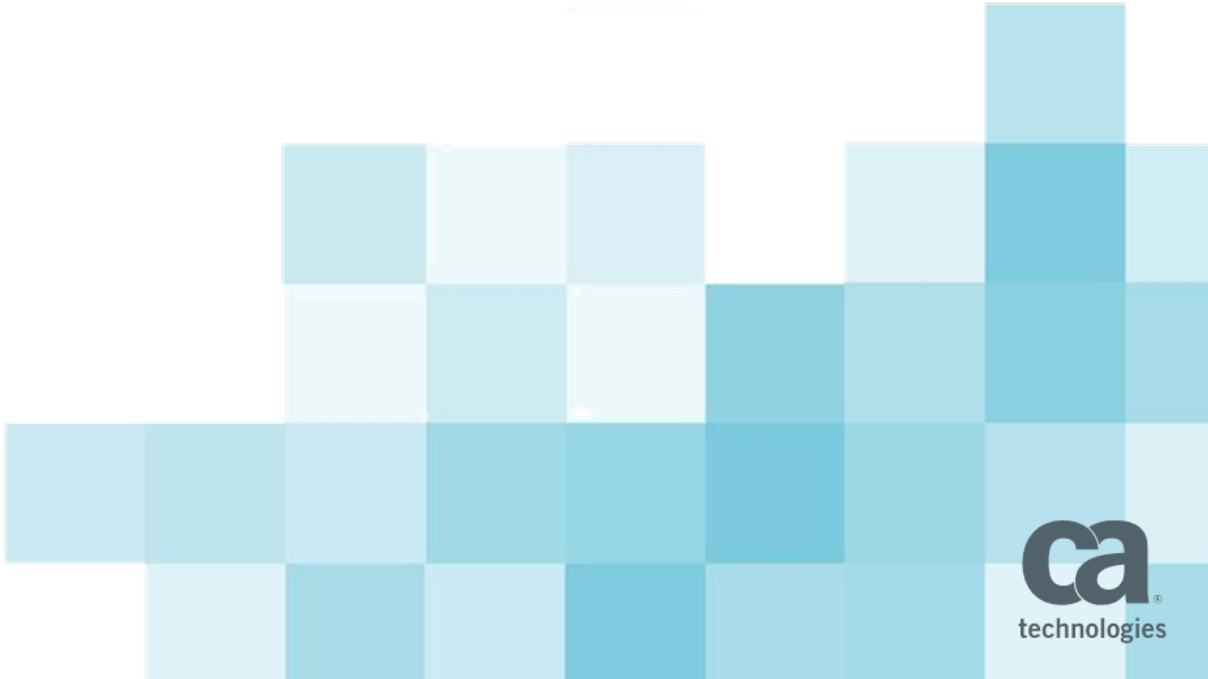
APPLICATION

INTEGRATION LAYER

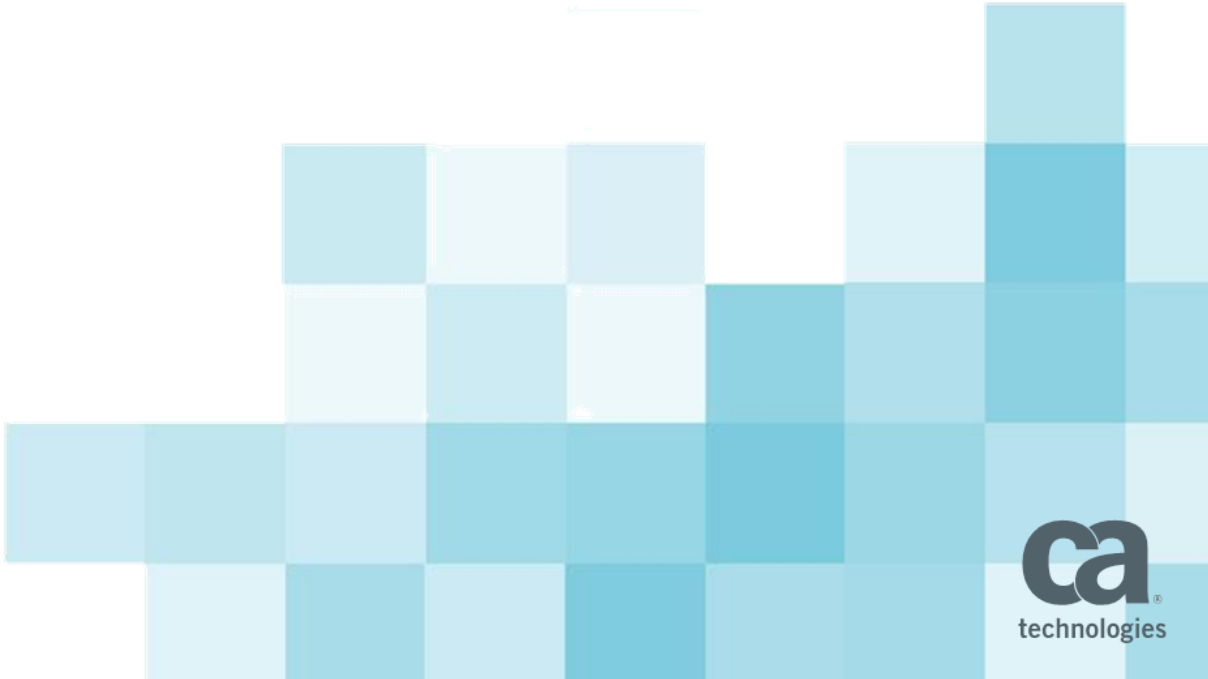
BACKEND

ca technologies

What is the story of Service Virtualization?

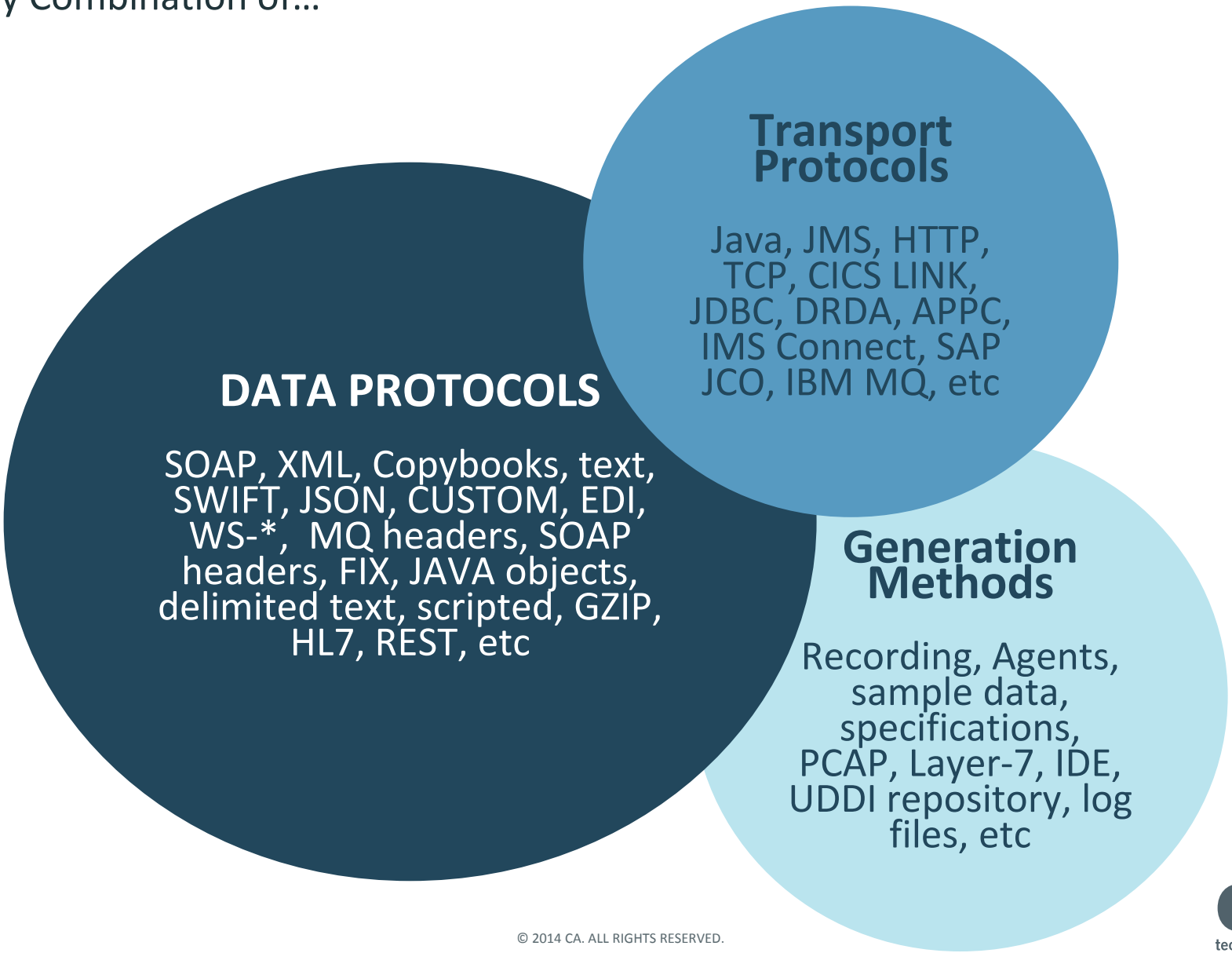


What can we Virtualize?

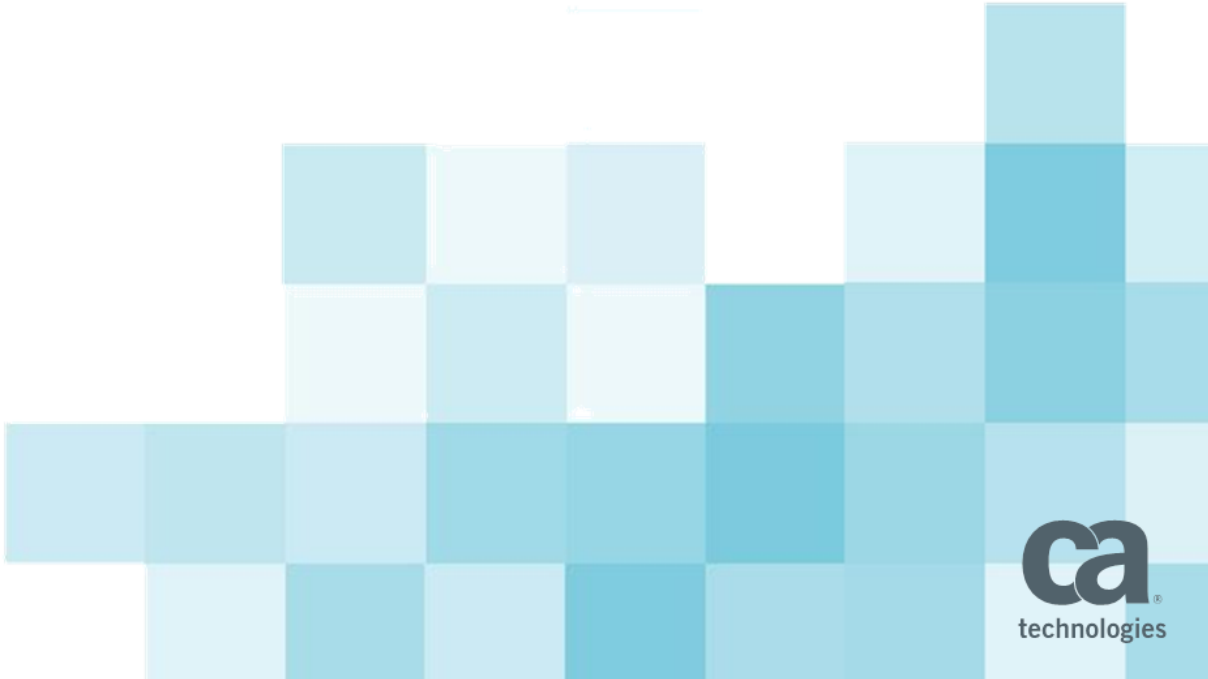


Lisa Virtualizing Capabilities

Any Combination of…

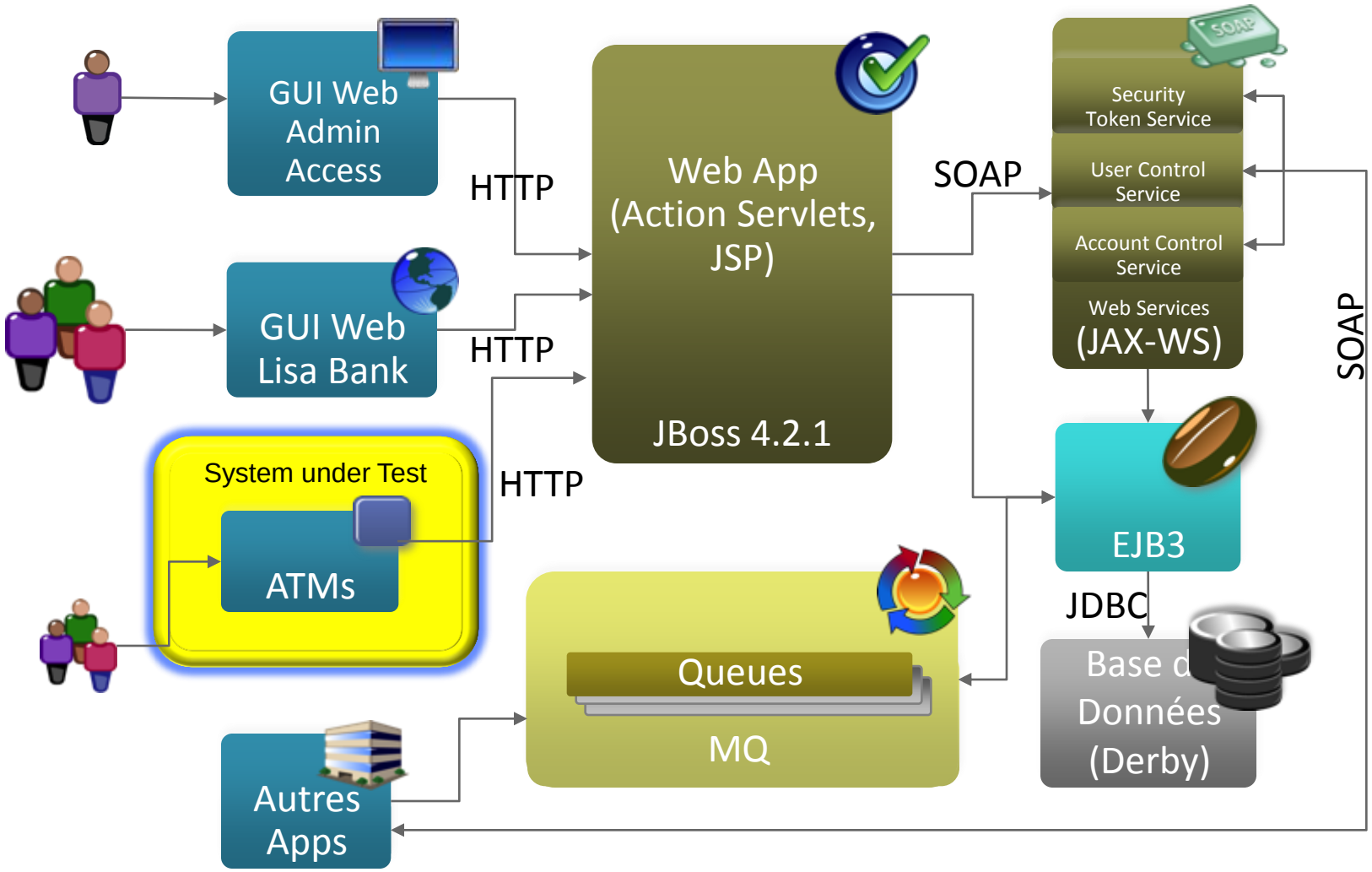


Well, stop babbling and SHOW ME

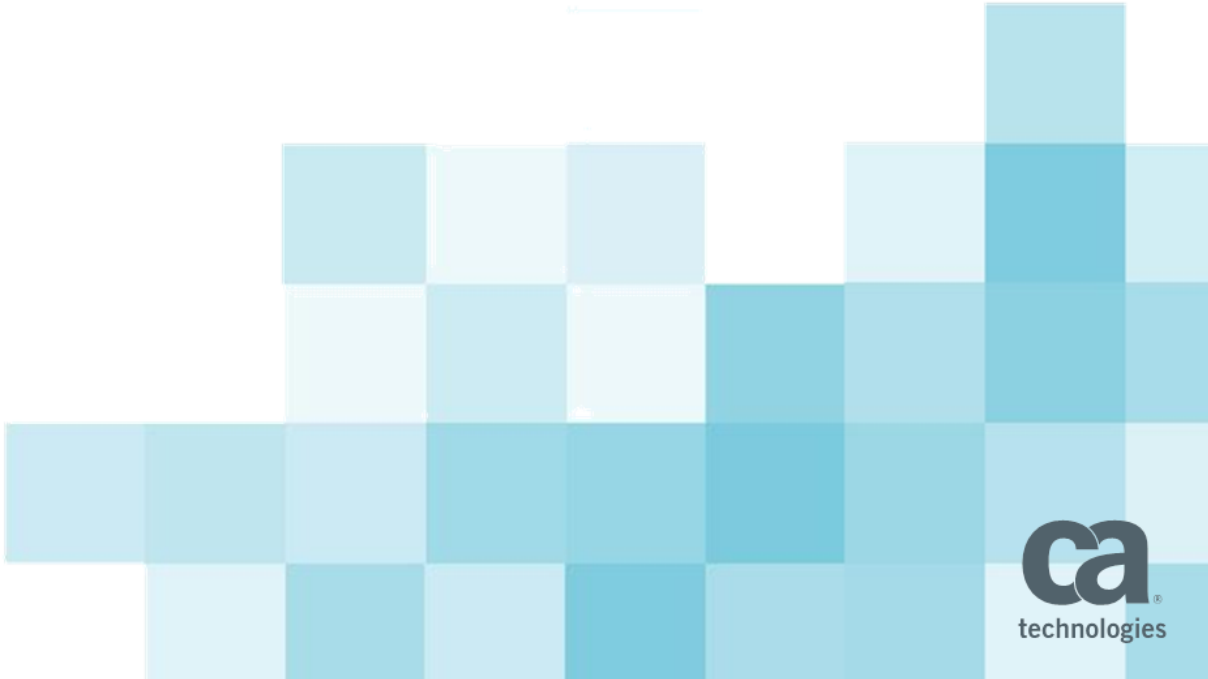


ca technologies

Lisa Bank application and its constraints



Who does it?



Capability	Benefit	CA LISA® Service Virtualization	Easy Mock/Mockito
Reduce dependencies and constraints in application development	Reduce cycle times, earlier defect discovery, more efficient use of resources		
Virtualize test data and provide rich data for comprehensive test paths	Faster setup/teardown, greater stability for test automation		Test data must be designed for specific use cases
Enable high-performance environments	Better realism and quantity of performance testing (Thread Safe)		Not designed for performance testing
Complete testing of application business logic	Eliminate mocking from "inside" the application that results in cutting off business logic		Inside Application Code
Provide reuse and collaboration across development teams	Virtualize a Service once and reuse across development teams and applications		Every app requires writing mocks that could be shared
Incorporate into Continuous Integration Process	Improve the quality of software and reduce delivery time. Hook into all stages of CI (Dev, QA, Performance, Integration and Clone Test) environments		Limited outside of Dev Environment
Reduce time and cost of Maintenance for dev and test environments	Increase developers time to focus on application code vs. maintaining mocks and test data that are code dependent		Code Dependent
Expansive Code Coverage	Enable a consistent and standardized SDLC / CI Process with solution that supports all code bases		Java Only
UI to modify request/response	Resources with no developer skills can create and update models		Coding required
Bulk import/recording of transactions	Leverage large volumes of actual behavior and data		No recording or importing of transactions

Q & A?