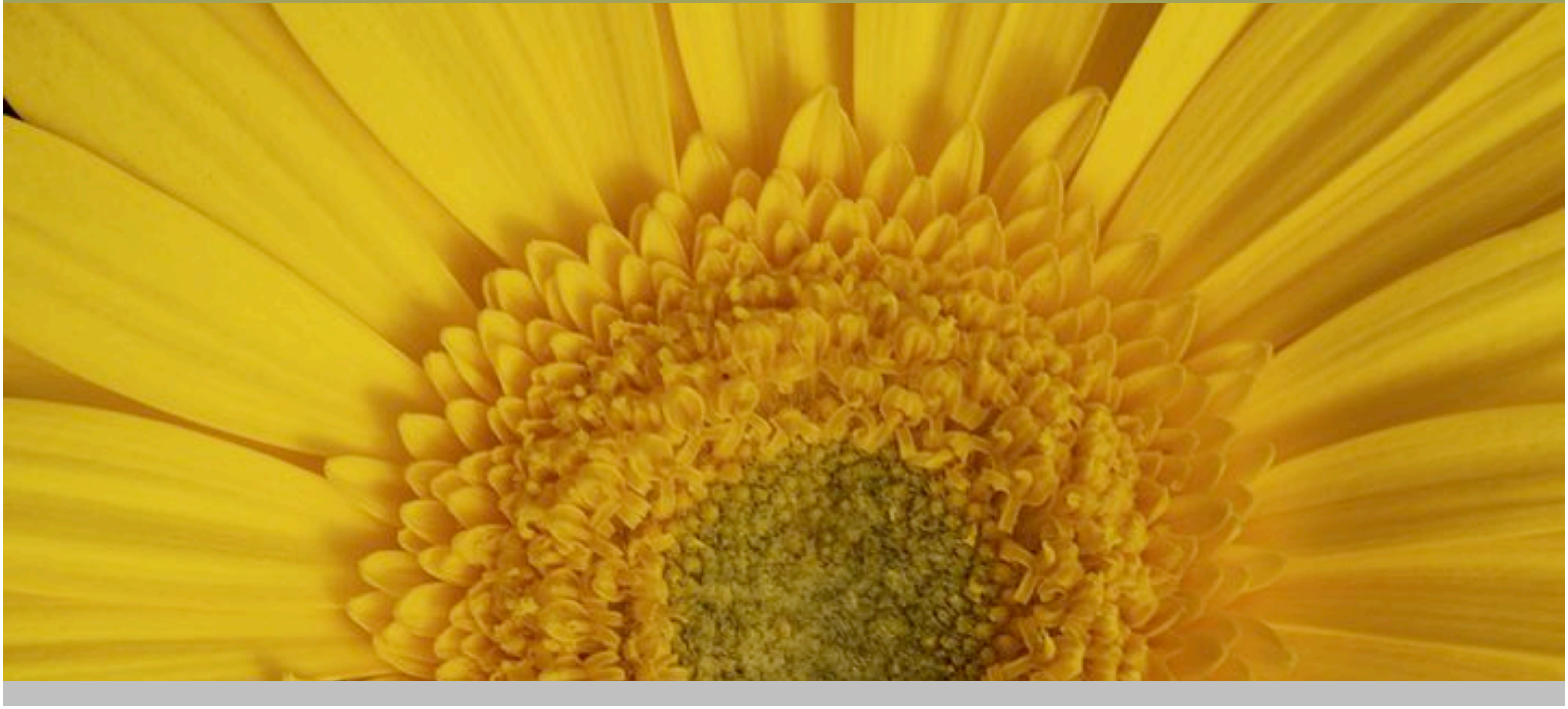
# For SOA, the future of Quality is Federated

Chris Kraus
Chris Dworkin
Steve Thomas
iTKO, Inc.
October 1, 2007

iTKO LISA™

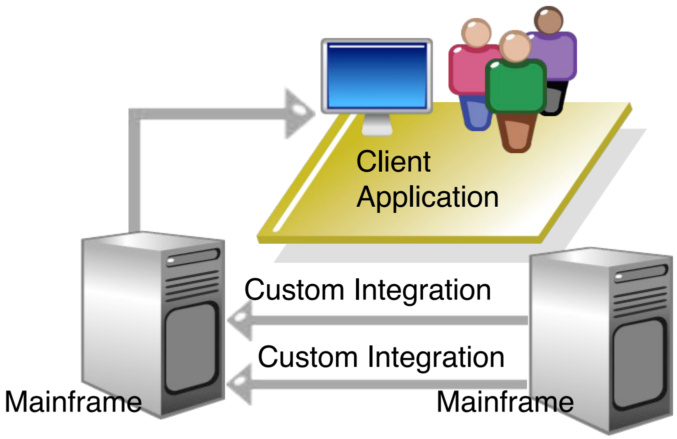
# Defining SOA and new challenges



iTKO LISA™

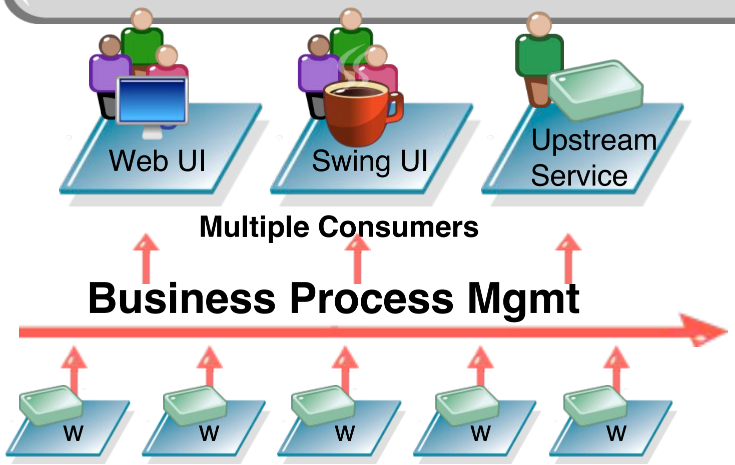
# What is SOA?

## Traditional Applications



- Rigid Infrastructure, Large Applications
- Long cycle (months) to reflect business changes
- Very expensive to maintain integration

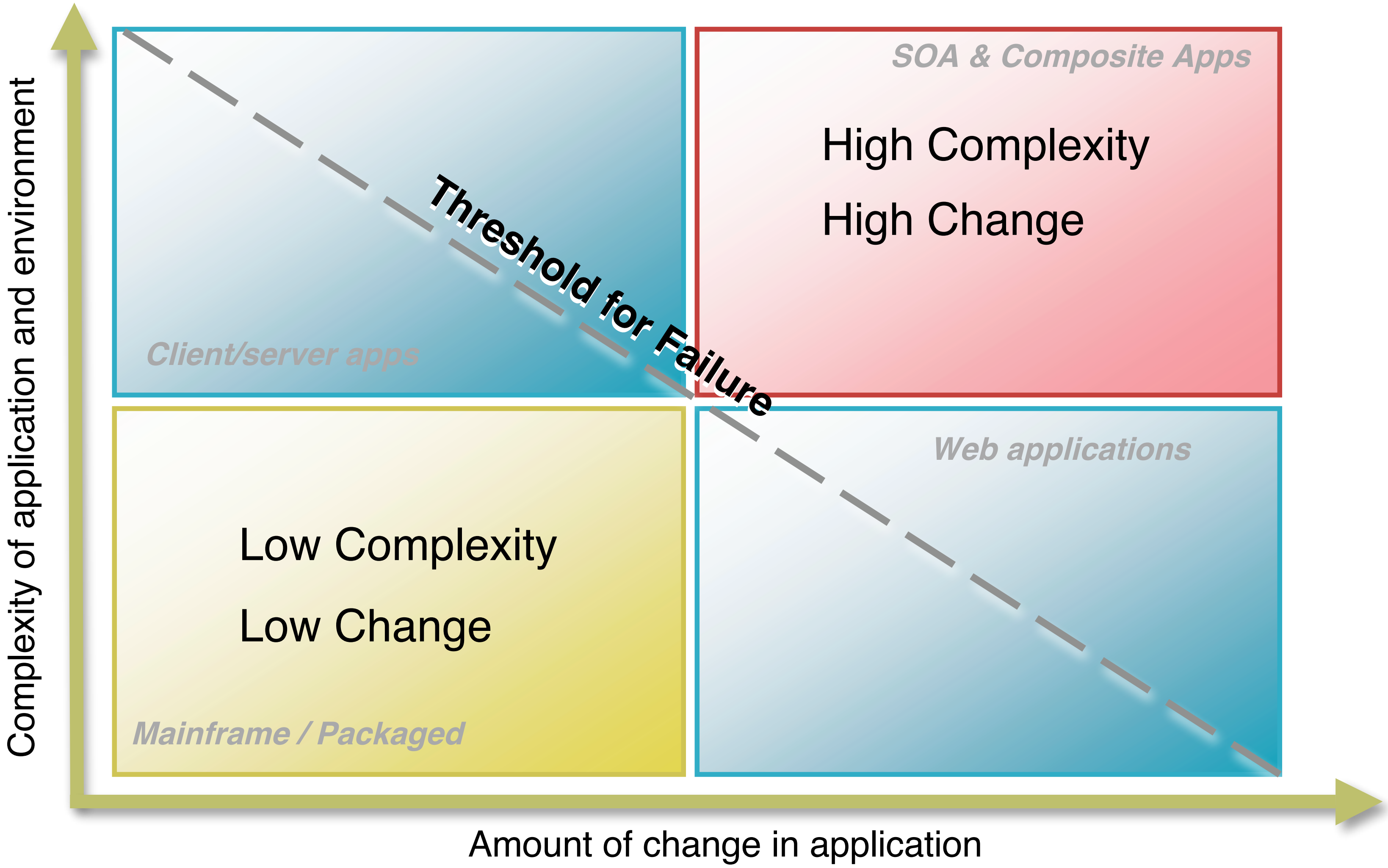
## SOA Approach



**Multiple Technologies Exposed As Services**

- Loosely-coupled & Heterogeneous integration strategy
- Leverage & Extend existing & 3$^{rd}$ Party Services
- Agile IT and Business Alignment

# Tipping Point for Automation



SOA & Composite Apps

High Complexity

High Change

Threshold for Failure

Client/server apps

Web applications

Low Complexity

Low Change

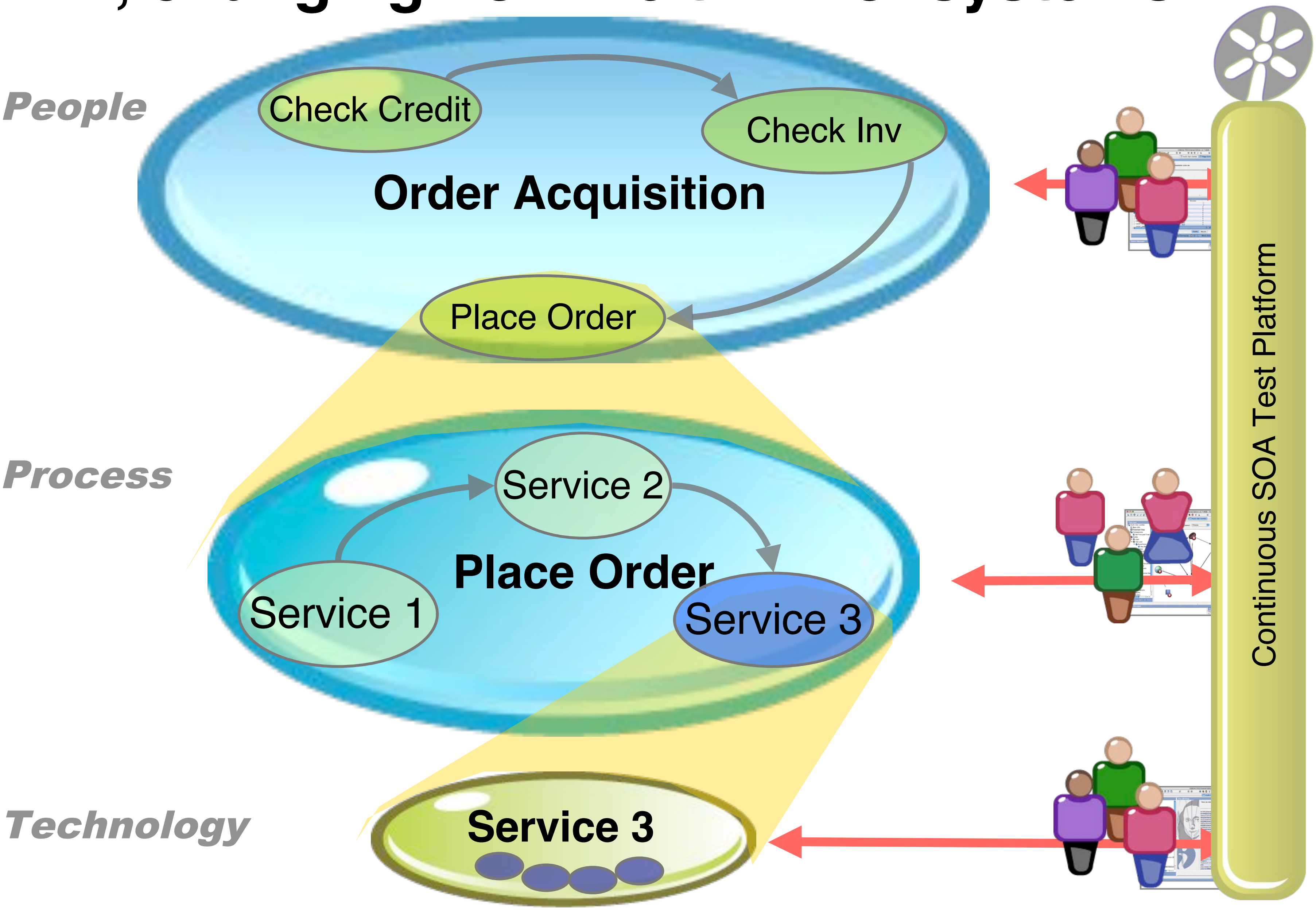
Mainframe / Packaged

Complexity of application and environment

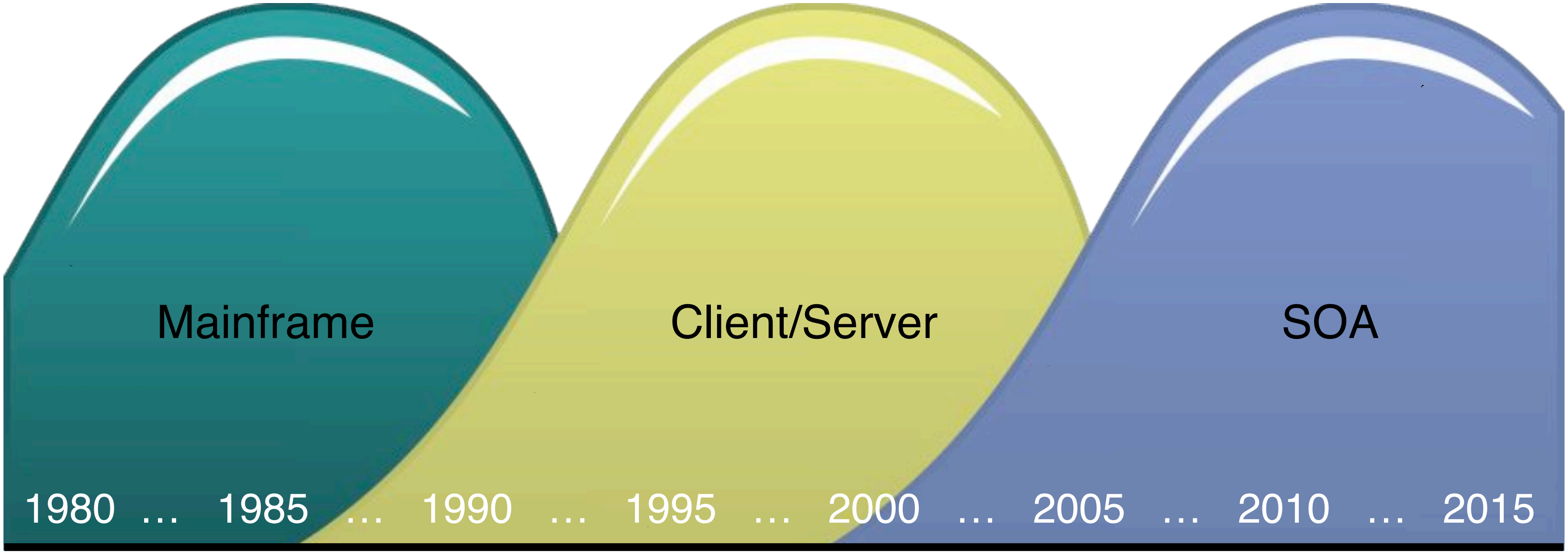
Amount of change in application

**↑ Complexity x ↑ Change = ↑ Risk of Failure**

# PPT; changing how we think of systems

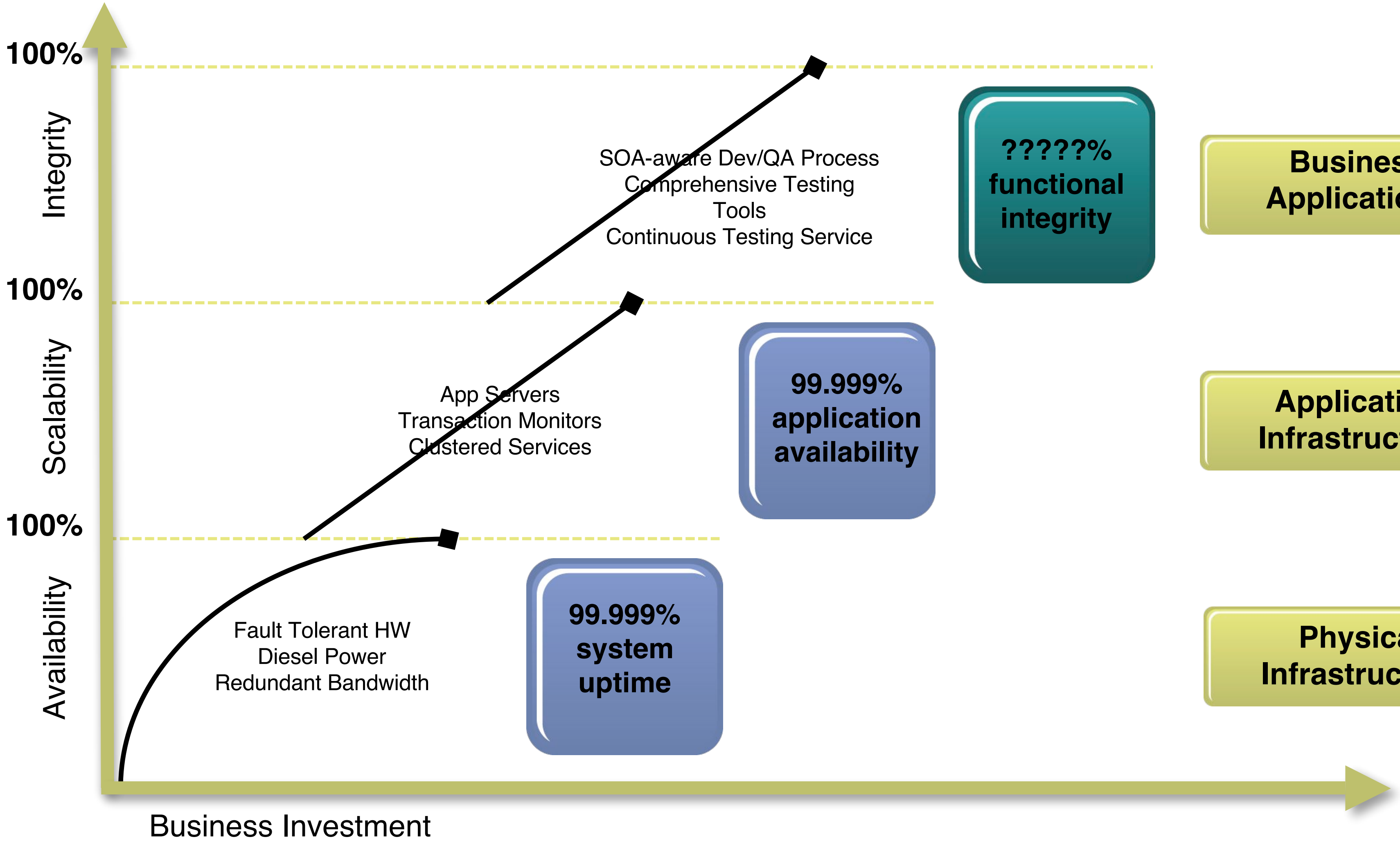


# Fundamental Enterprise IT Shift to SOA



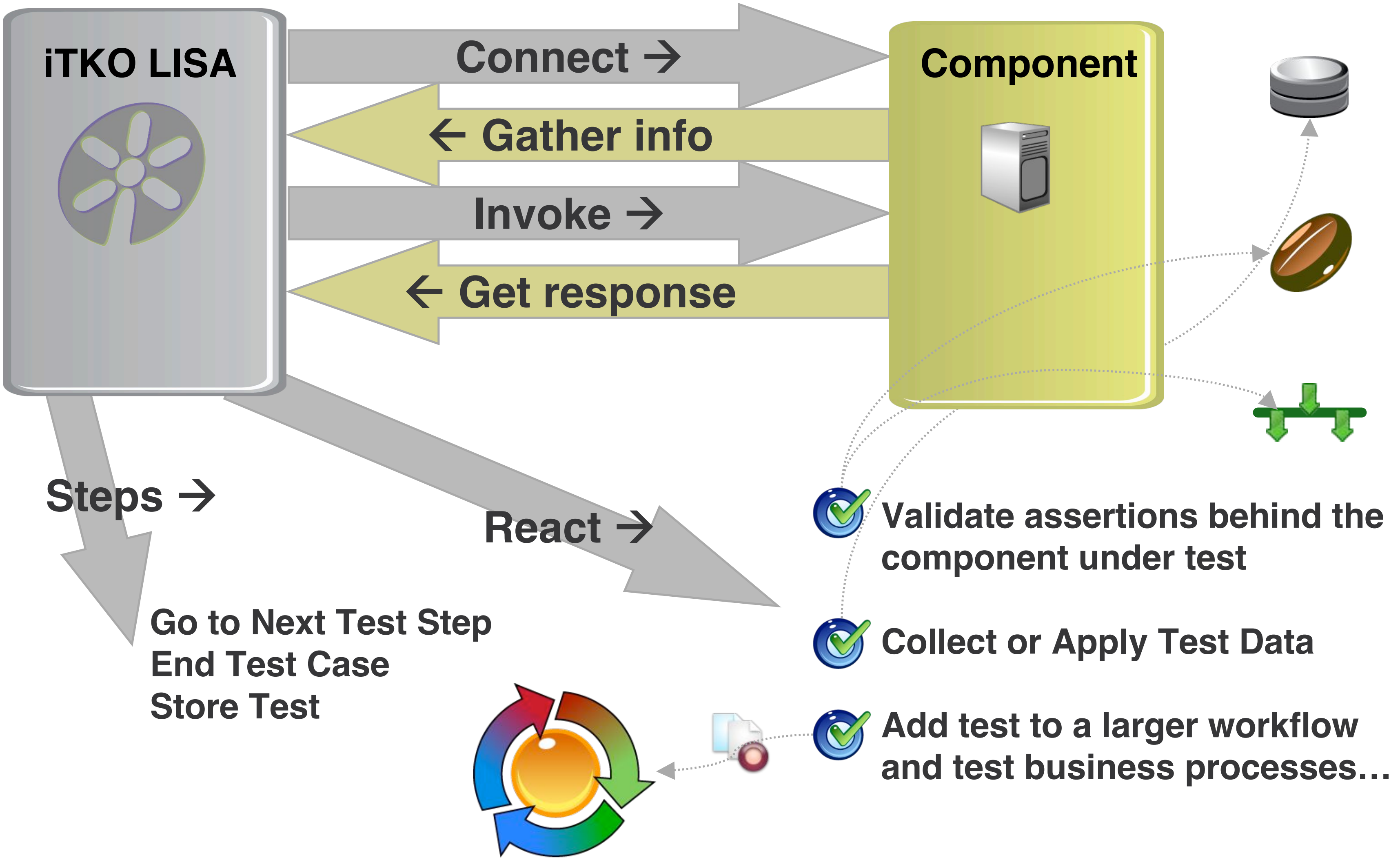
- Aberdeen: 90% of $1B+ (Large) companies have SOA In Process by 2006.
- Gartner: "By 2009, more than 80% of development and integration budgets will be dedicated to applications delivered as an SOA."
- McKinsey survey in Forbes: one of two CIO Priorities for 2007, 70% implementing SOA in 2007

# The Goal: SOA at 5-9's



**Availability to the business is the product of all three**

# How Tests of Components differs



iTKO LISA

Connect →

← Gather info

Invoke →

← Get response

Component

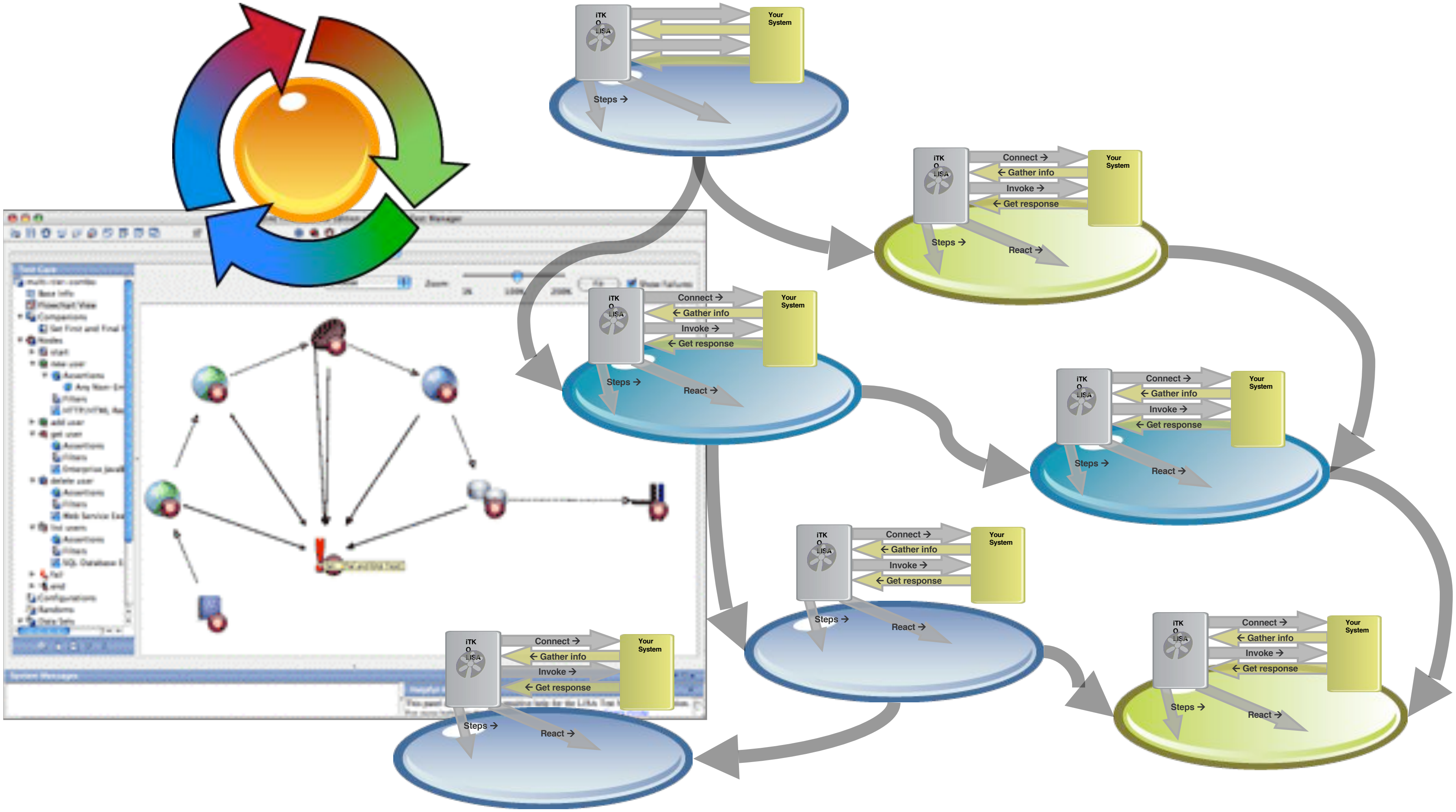
Steps →

React →

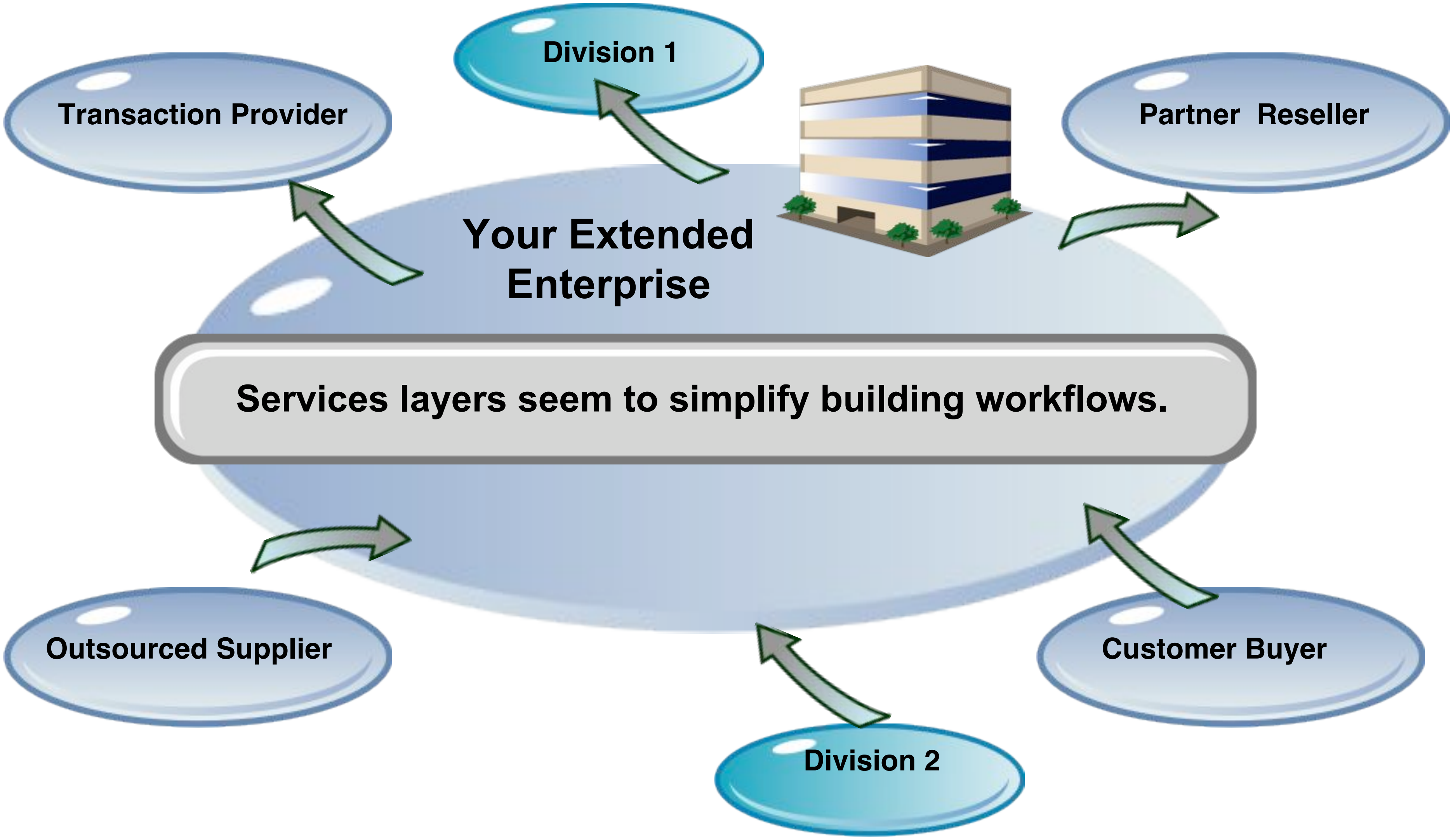
Go to Next Test Step
End Test Case
Store Test

- Validate assertions behind the component under test
- Collect or Apply Test Data
- Add test to a larger workflow and test business processes…

# Combine Tests as a Workflow

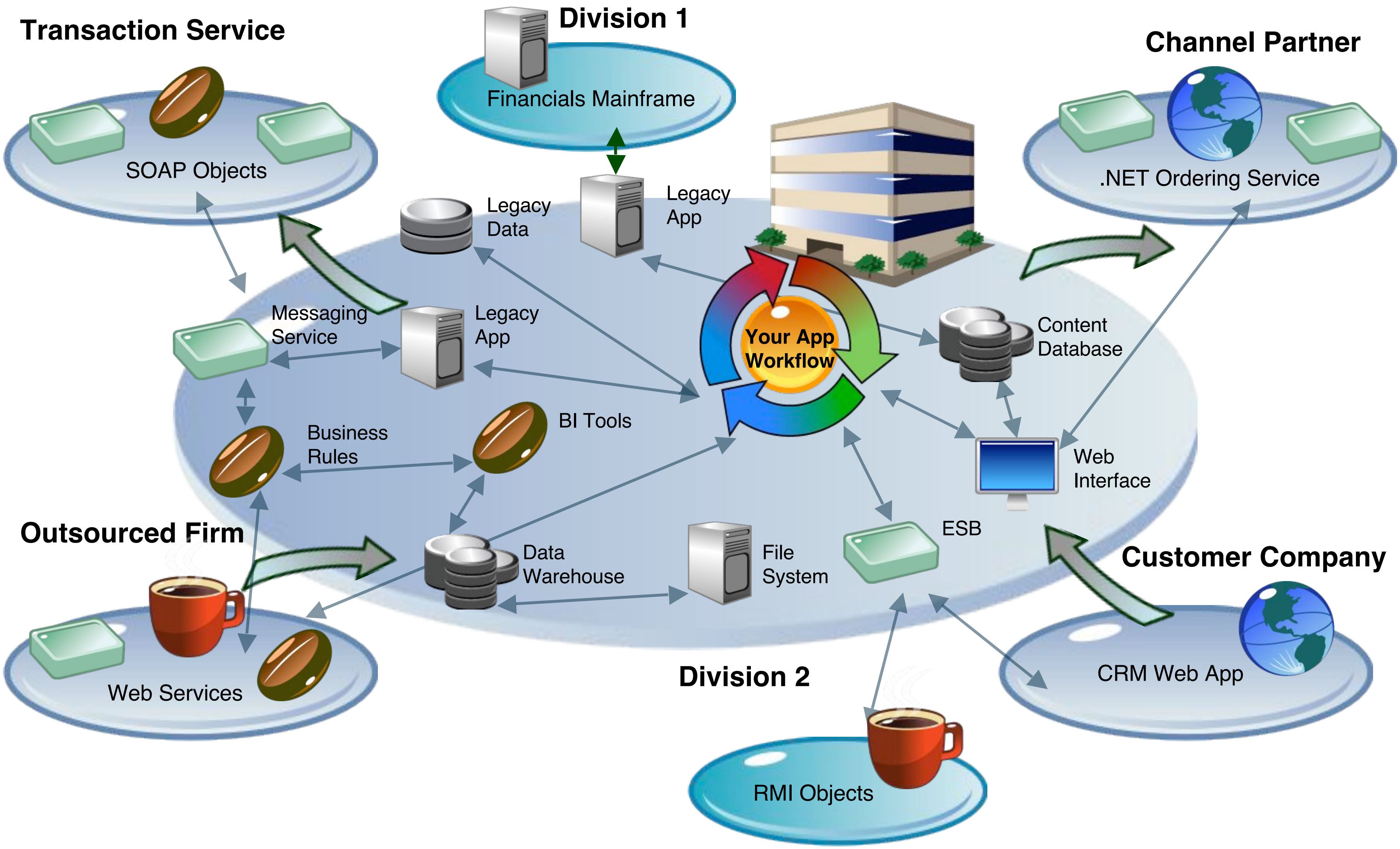


# Challenges of a service-based approach

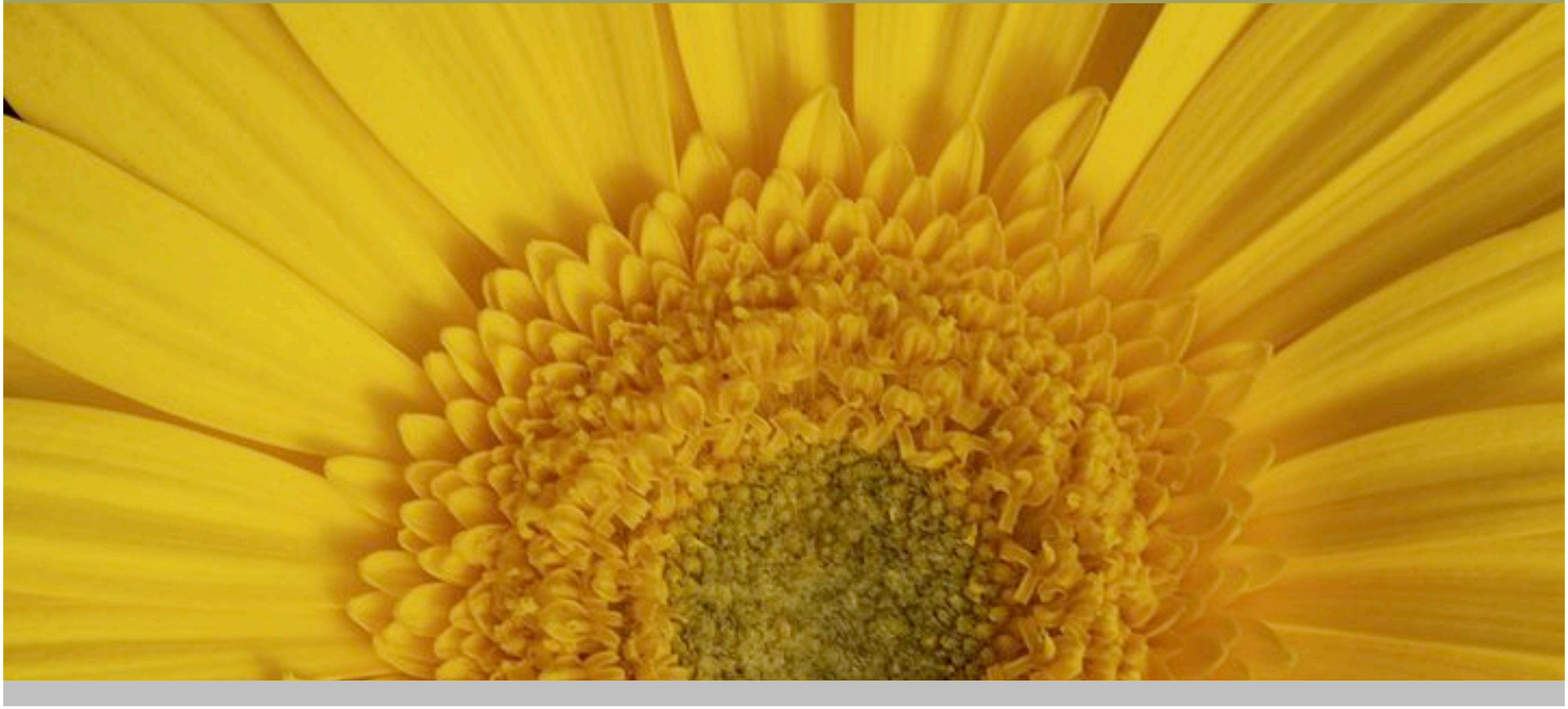


Services layers seem to simplify building workflows.

# But services layers hide complexity

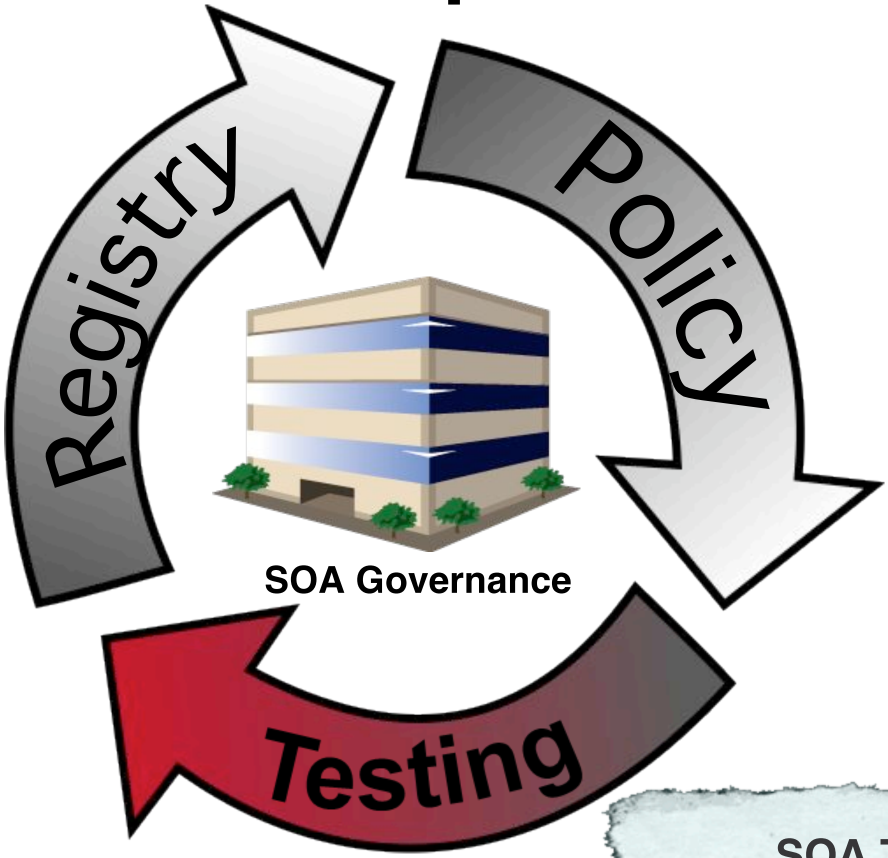


# What is Governance of SOA



iTKO LISA™

# Three Components of SOA Governance

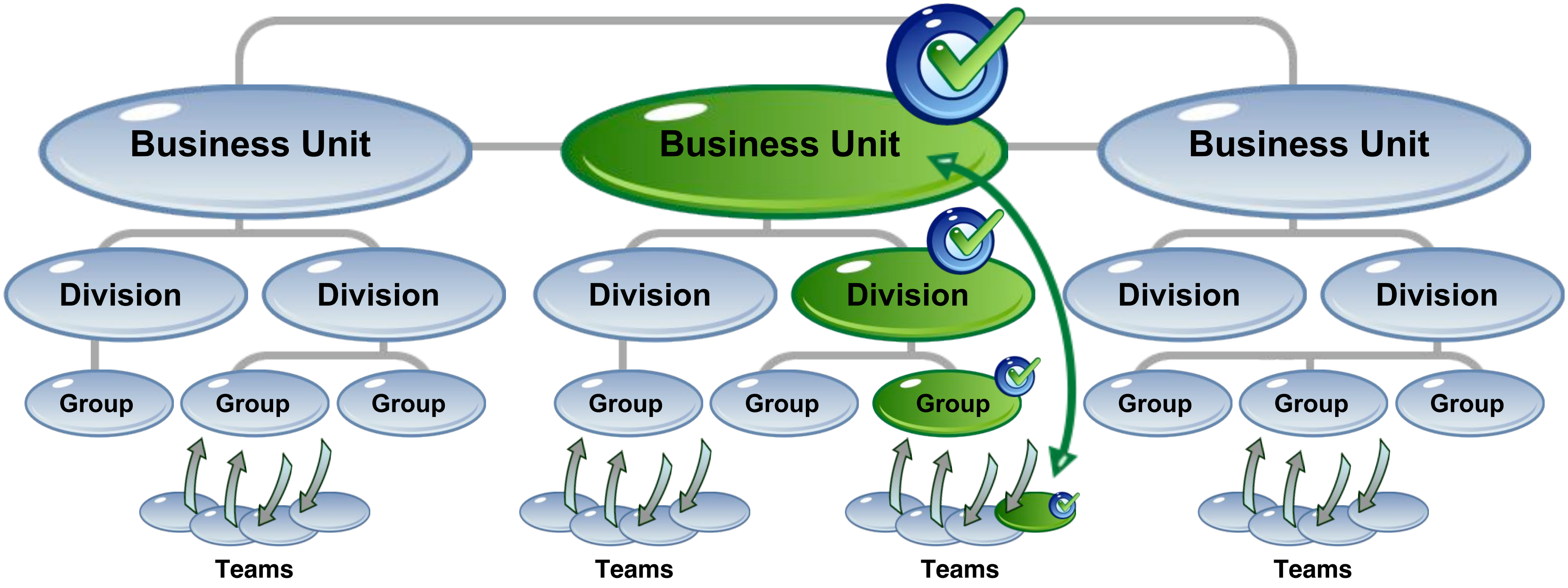


- **Gartner's Frank Kenney:** *SOA Governance consists of Registry, Policy, and SOA Testing*
- **Gartner "Hype Cycle" 2006:** *SOA Testing the only "transformational" value change in Application Development practices*
- **Hurwitz Group 2007 predictions:** *"Quality and software testing, a longtime stepchild of the software market, will get hot. This will be especially true in the context of ensuring that business services perform the functions they are designed for in a predictable manner."*

### SOA Testing: the #1 Prediction for 2007

"As a result of this heightened awareness of the real challenges in maintaining a SOA implementation, **demand for SOA quality and testing solutions will skyrocket in 2007, leading to greater acquisitions, increased consolidation, new venture creation, and boatloads of case studies on the topic.**" - Jason Bloomberg, Senior Analyst, ZapThink

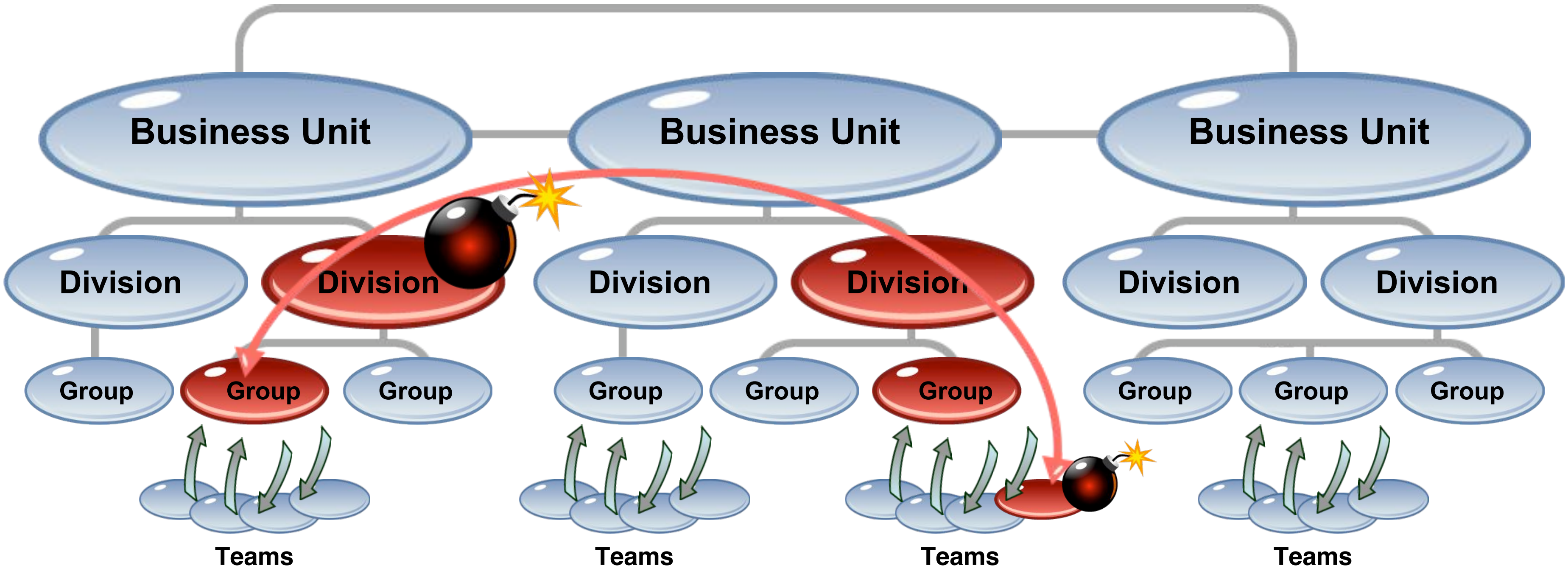
# Vertical Governance = Piece of Cake



For vertical hierarchies within organizations, there is an expected level of shared trust.

- Higher level can expect the underlying development team to “build to order” IT functionality for it
- Service development team can expect the consumer to leverage the service according to understood business requirements

# Horizontal Service Reuse = Anarchy

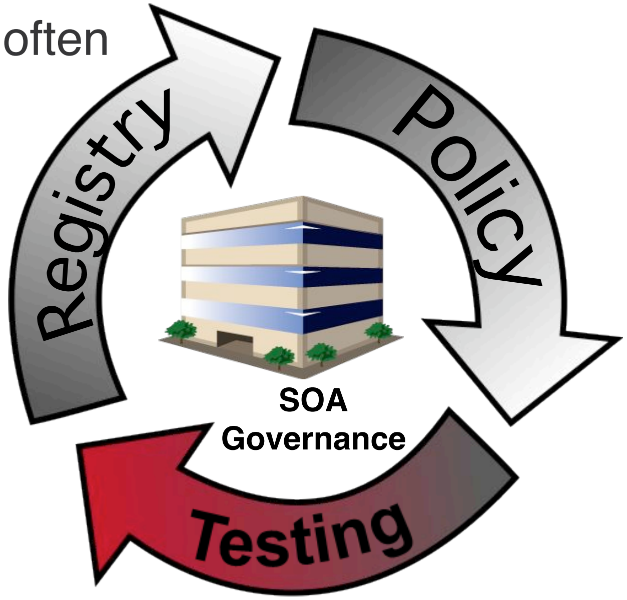


Across different business units, coordinating the use of a service is difficult.

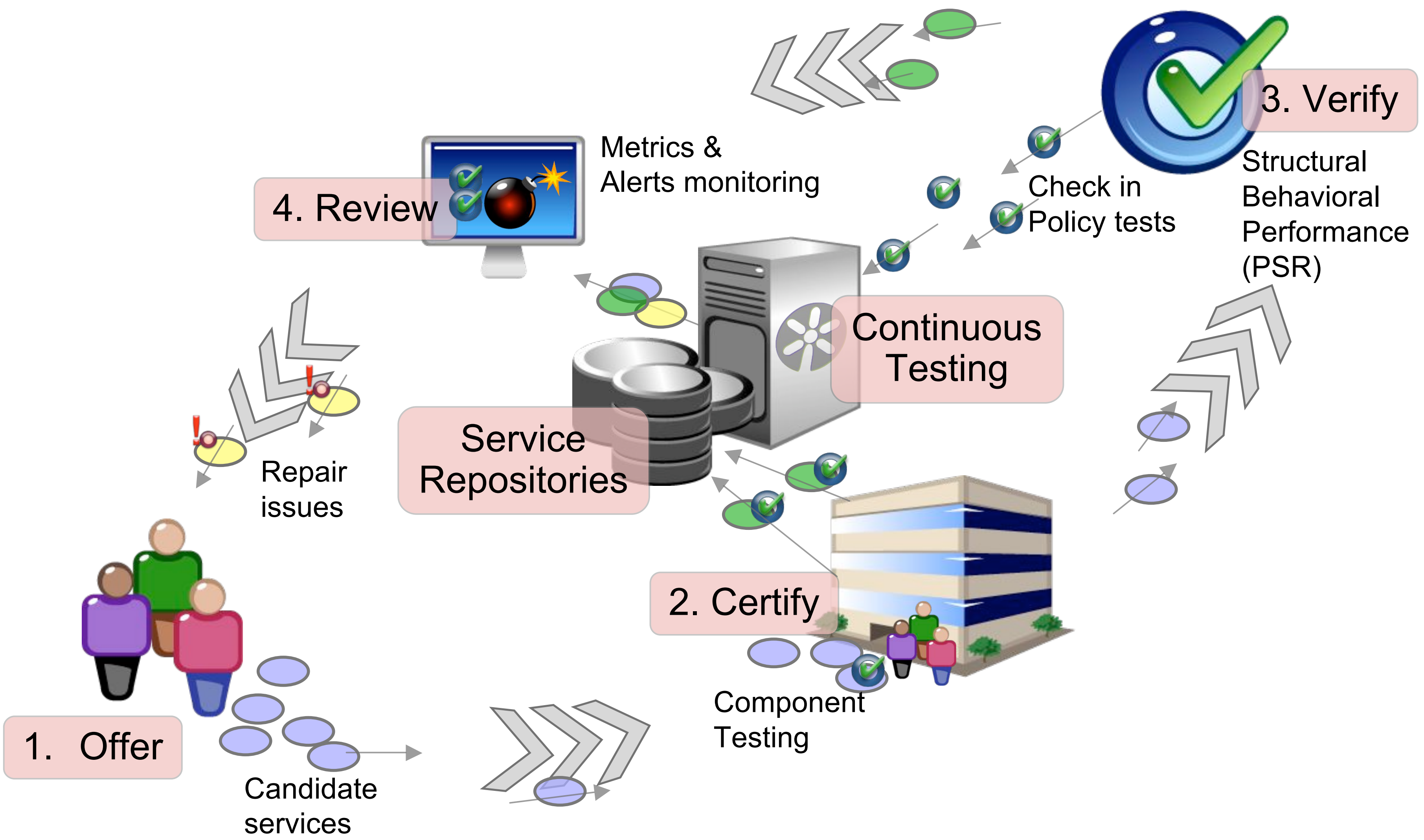
- Service Providers want to establish reuse of services, but they are answerable to different stakeholders
- Upstream Consumers of services do not provide clear enough Use Cases of how they will employ services
- Therefore teams build redundant functionality

# Governance Provides Trust

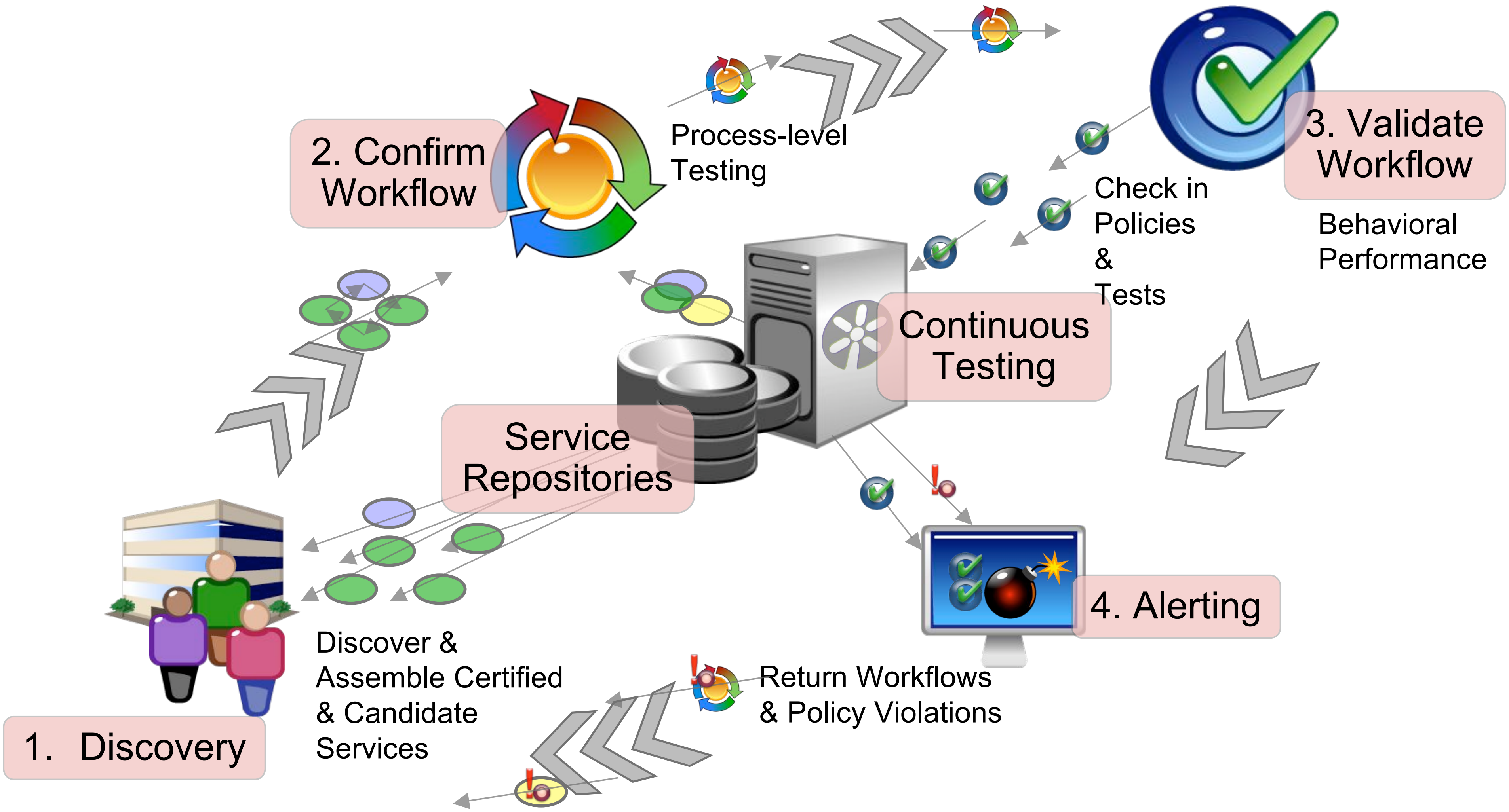
- Governance solutions must provide trust & transparency
- Key function is SOA Policy definition, modeling, and enforcement
- Three kinds of policy can be automated
  - **Structural** -- compliance to standards -- ‘the pin-outs’
  - **Behavioral** -- honoring functional expectations
  - **Performance** -- honoring performance / reliability expectations
- Critical to establish an environment that provides this, often federated across the organization(s)
  - Top level SOA COE built from participants
  - Clearly defined Publish and Consume Processes
  - Visibility to clearly identify root causes via reporting, alerting, dashboards
- We call this SOA Testing



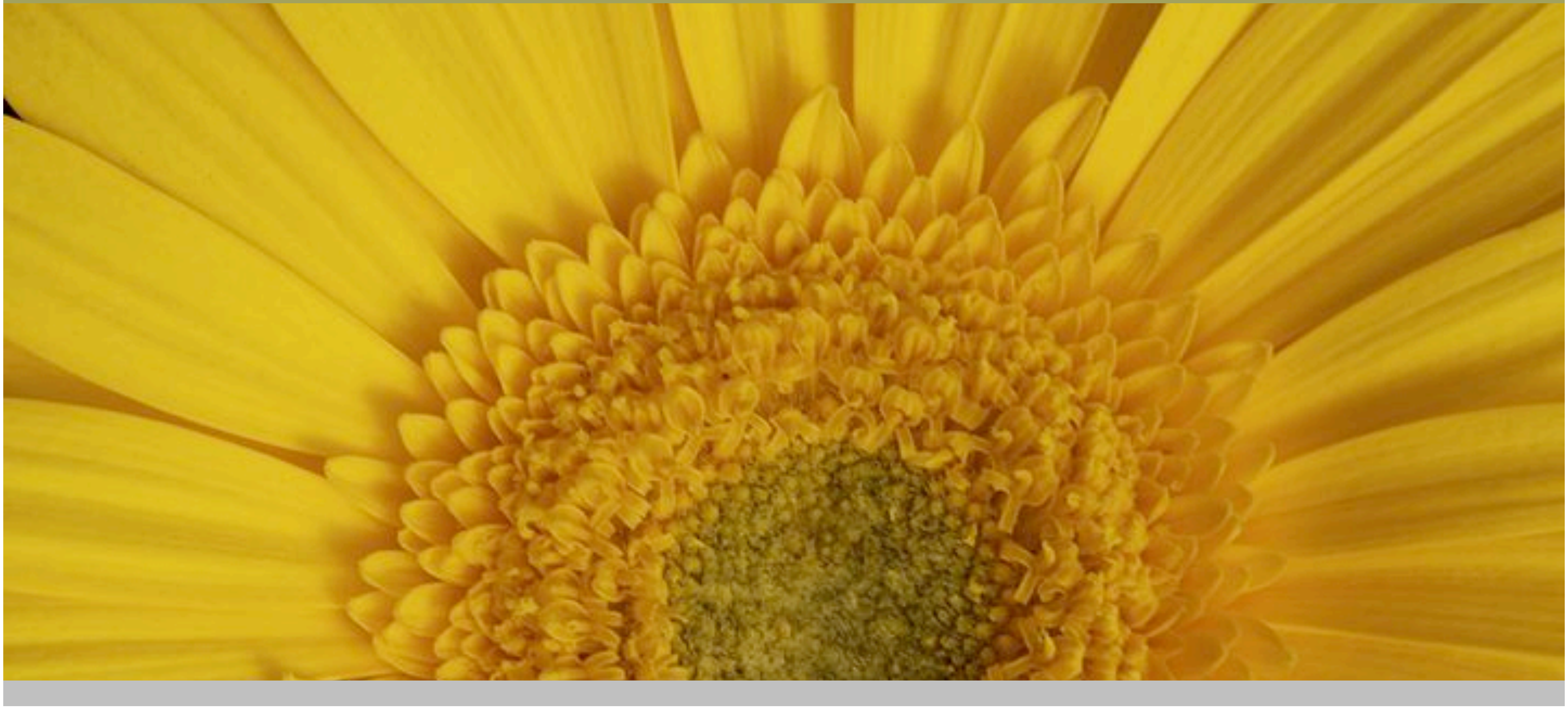
# SOA "Publish" Cycle



# SOA "Consume" Cycle

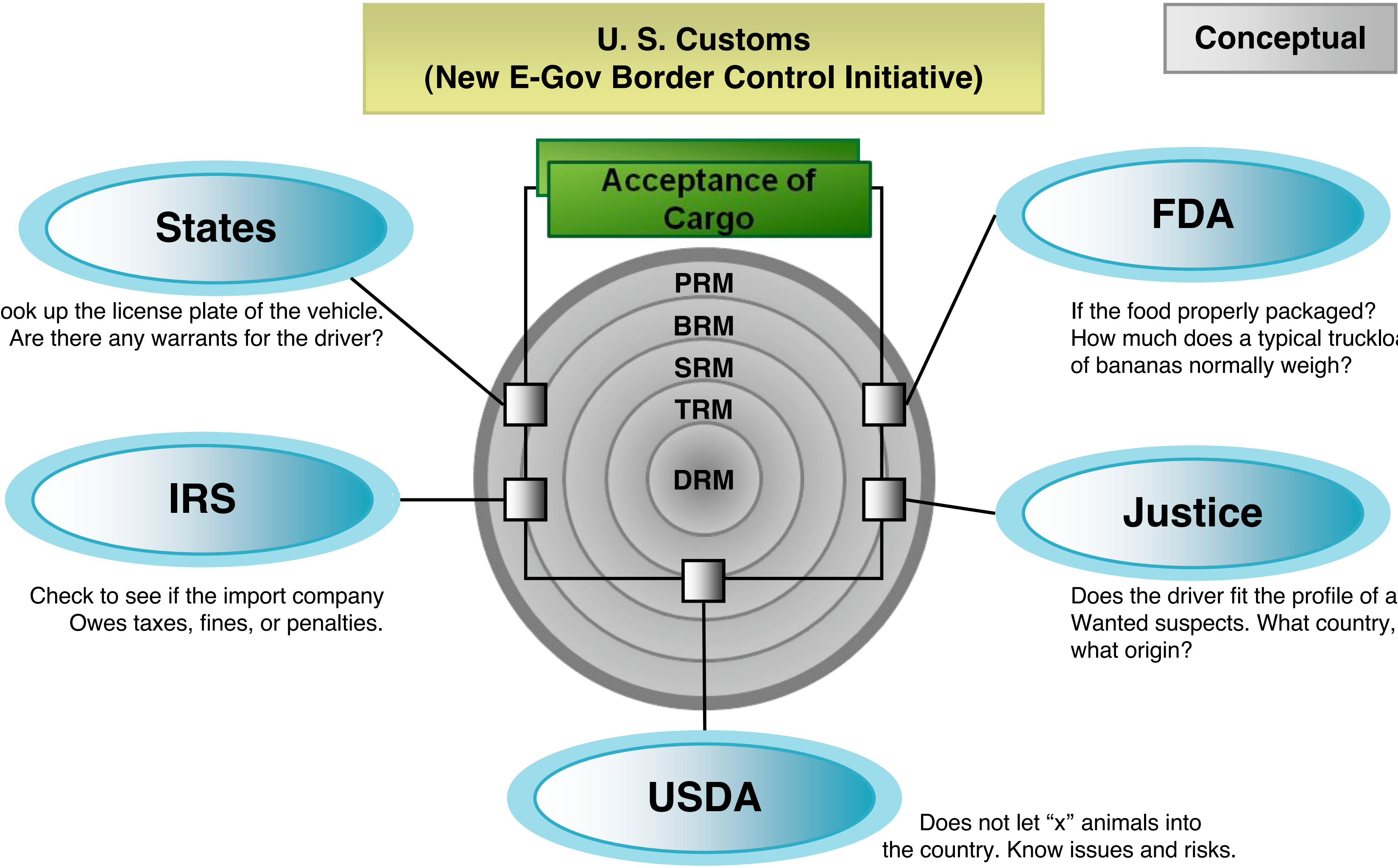


# Governance in Federal Government



iTKO LISA™

# FEA Example – Cross Agency



Look up the license plate of the vehicle.
Are there any warrants for the driver?

Check to see if the import company
Owes taxes, fines, or penalties.

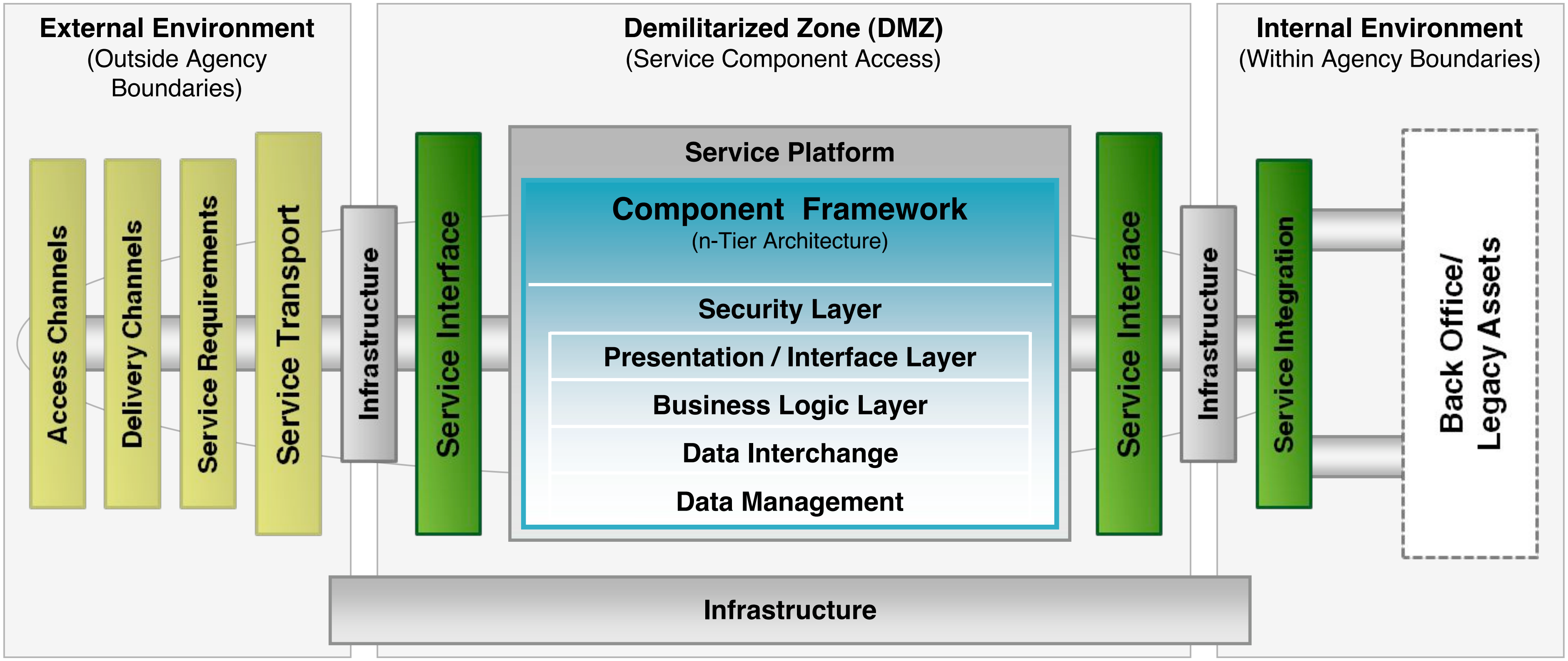
If the food properly packaged?
How much does a typical truckload
of bananas normally weigh?

Does the driver fit the profile of any
Wanted suspects. What country,
what origin?

Does not let “x” animals into
the country. Know issues and risks.

🟦 Agency EA ▨ Service Component ⬛ E-Gov Architecture Guidance ▨ Federal Enterprise Architecture

# “Complete” Example – FEA TRM



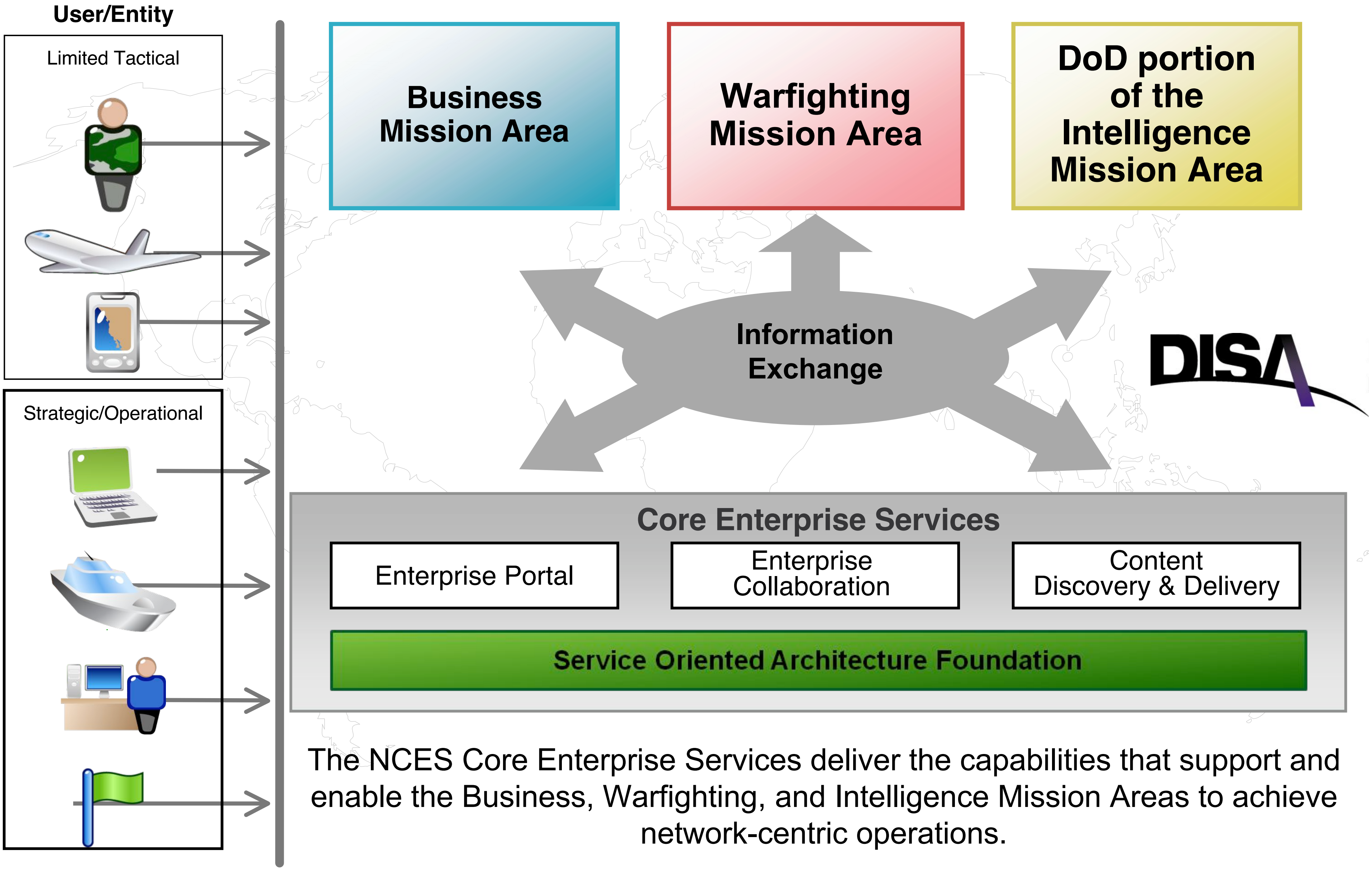
🟨 Service Access and Delivery

⬜ Service Platforms and Infrastructure

🟦 Component Framework

🟩 Service Interface And Integration

# Defense Example - NCES Increment I



User/Entity

Limited Tactical

Strategic/Operational

Business Mission Area

Warfighting Mission Area

DoD portion of the Intelligence Mission Area

Information Exchange

Core Enterprise Services

Enterprise Portal

Enterprise Collaboration

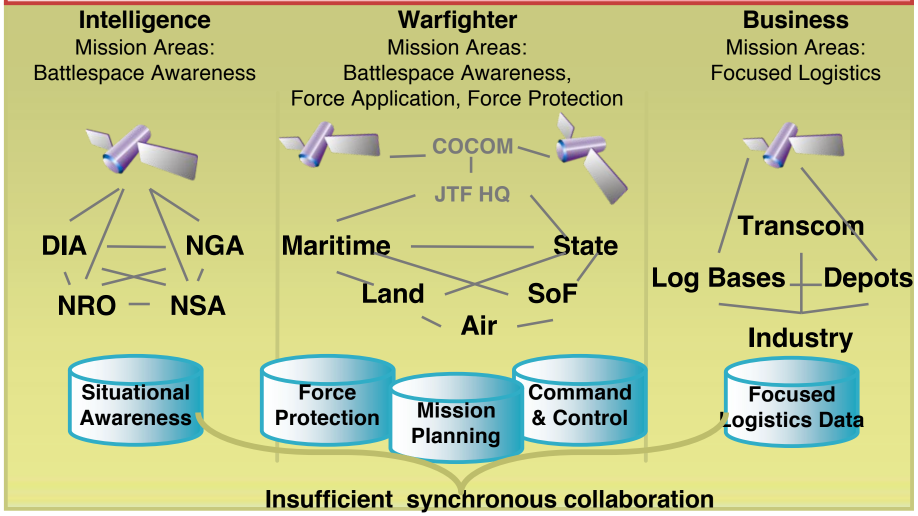
Content Discovery & Delivery

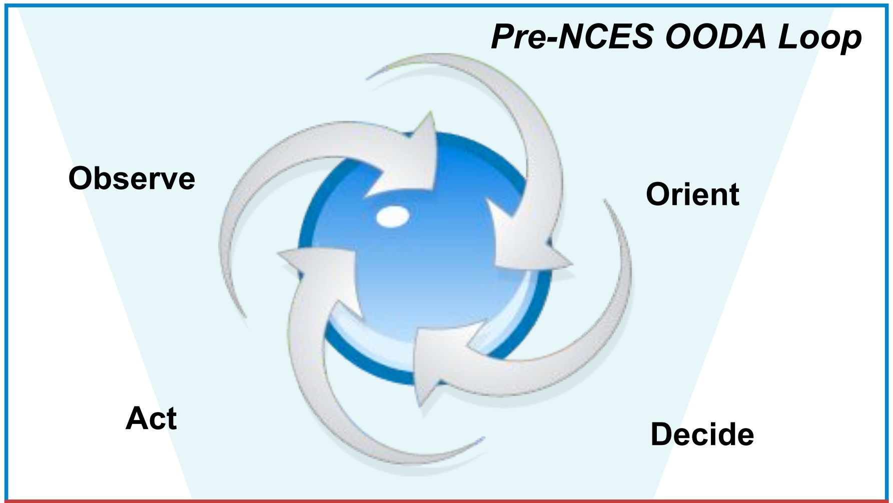
Service Oriented Architecture Foundation

The NCES Core Enterprise Services deliver the capabilities that support and enable the Business, Warfighting, and Intelligence Mission Areas to achieve network-centric operations.

# Benefits to NCES

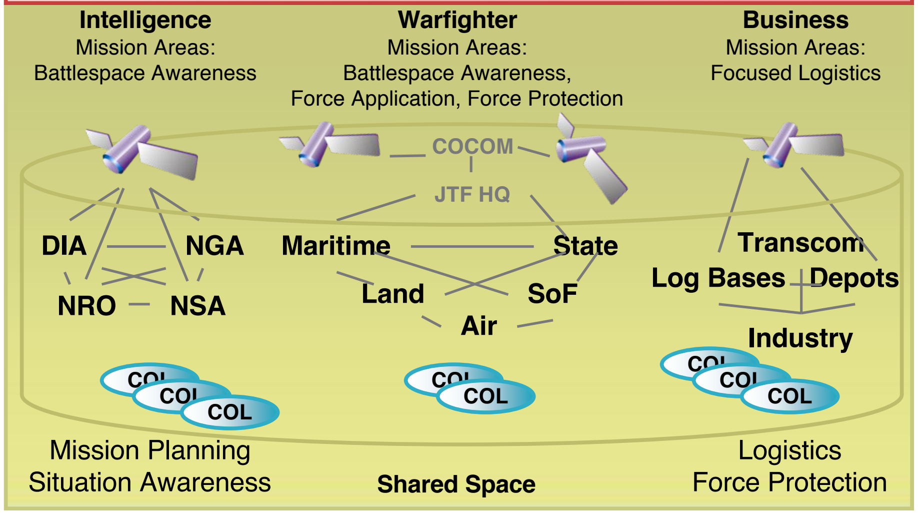
## "As-Is" Environment

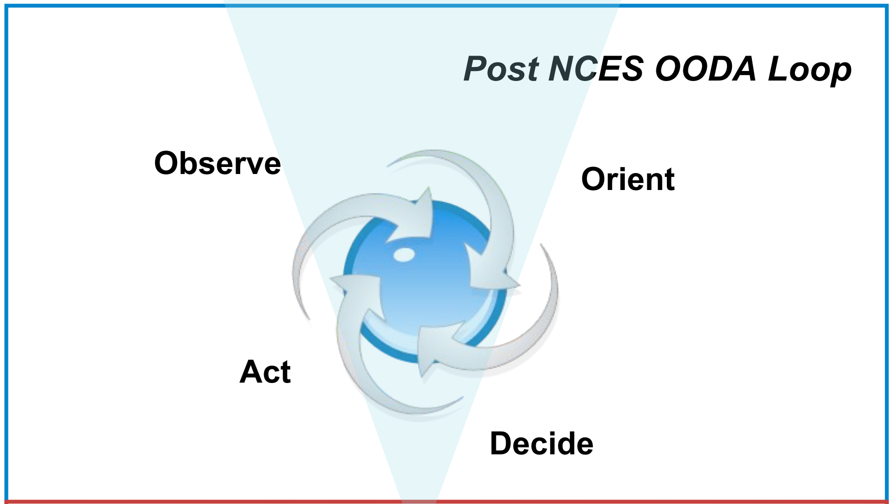




Stove-piped information = Large OODA Loop

## "To-Be" Environment





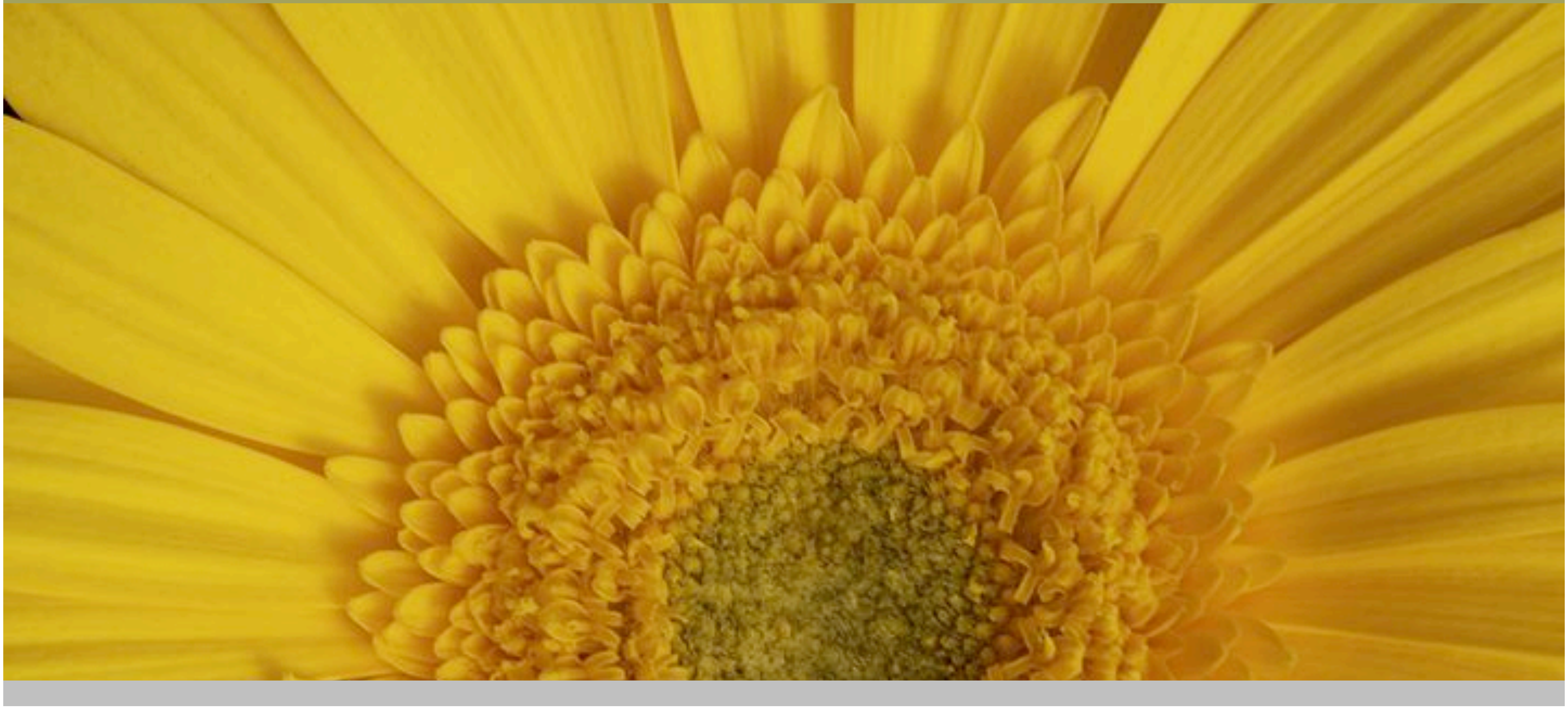
Shared, accessible information = Small OODA Loop

**NCES creates a shared information space, increasing decision velocity**

# It’s all about Trust

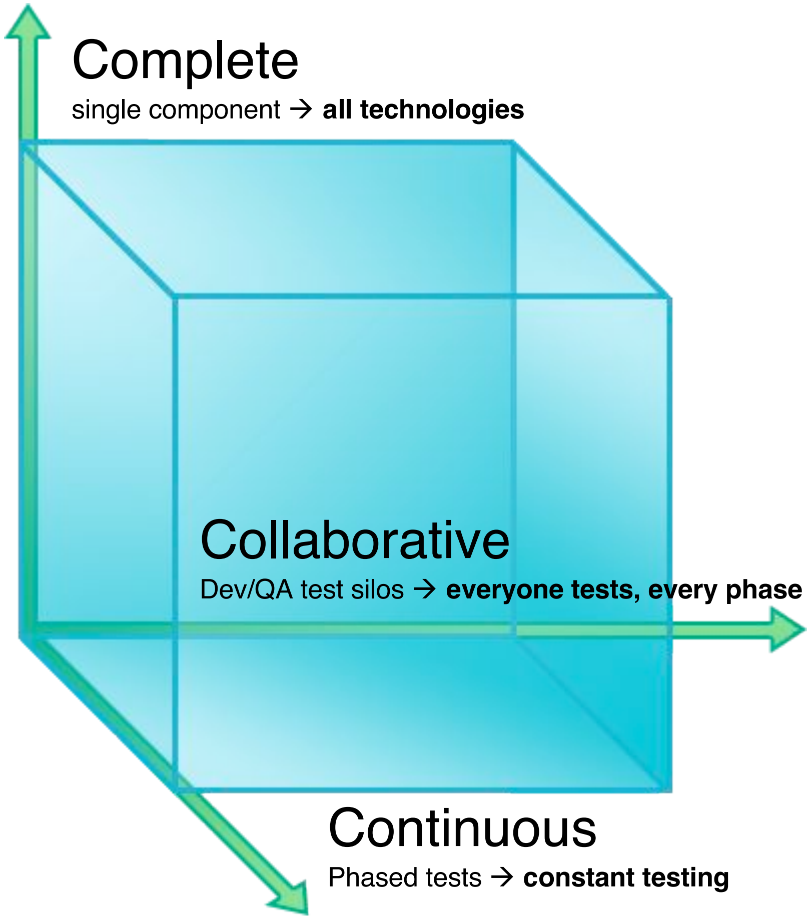
- Governance needs testing to provide visibility and enforcement of policy
- Visibility and enforcement allow service consumers to develop trust of services
- Trust leads to more frequent reuse and better utility of services
- Better utility and reuse, along with interoperability, are the base elements that allow services to be loosely coupled
- Loosely coupled services lead to better Business Agility

# Automation & Federation of Governance



iTKO LISA™

# Trust Requires: “Three C’s” of SOA Testing


Complete
single component → **all technologies**
Collaborative
Dev/QA test silos → **everyone tests, every phase**
Continuous
Phased tests → **constant testing**

## Complete Testing

- Every heterogeneous layer of architecture
- Test UI verify in system of record
- Reuse functional test for performance testing

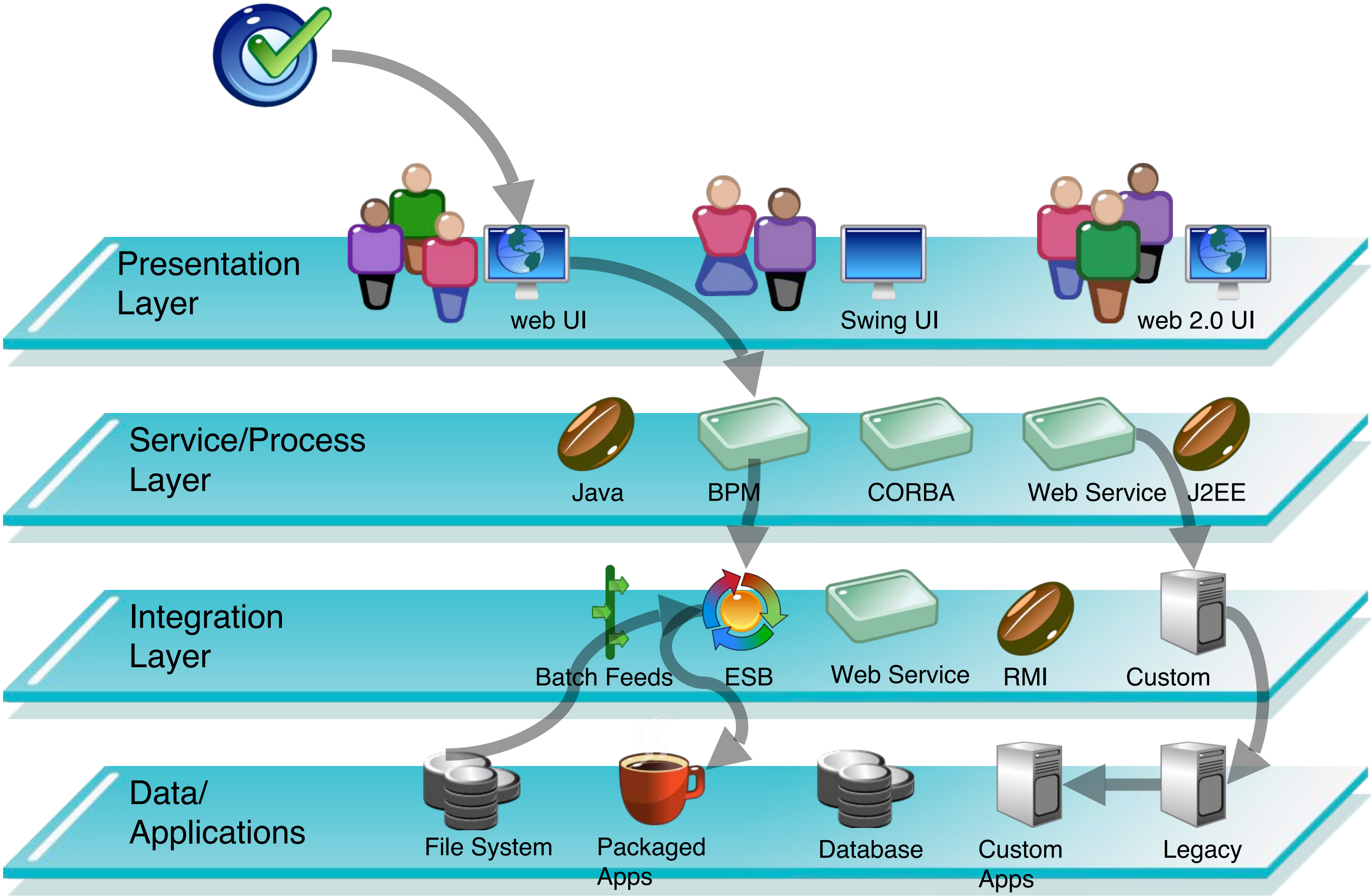
## Collaborative Testing

- Test early before UIs are created
- Not just dev, business analysts and QA should verify processes

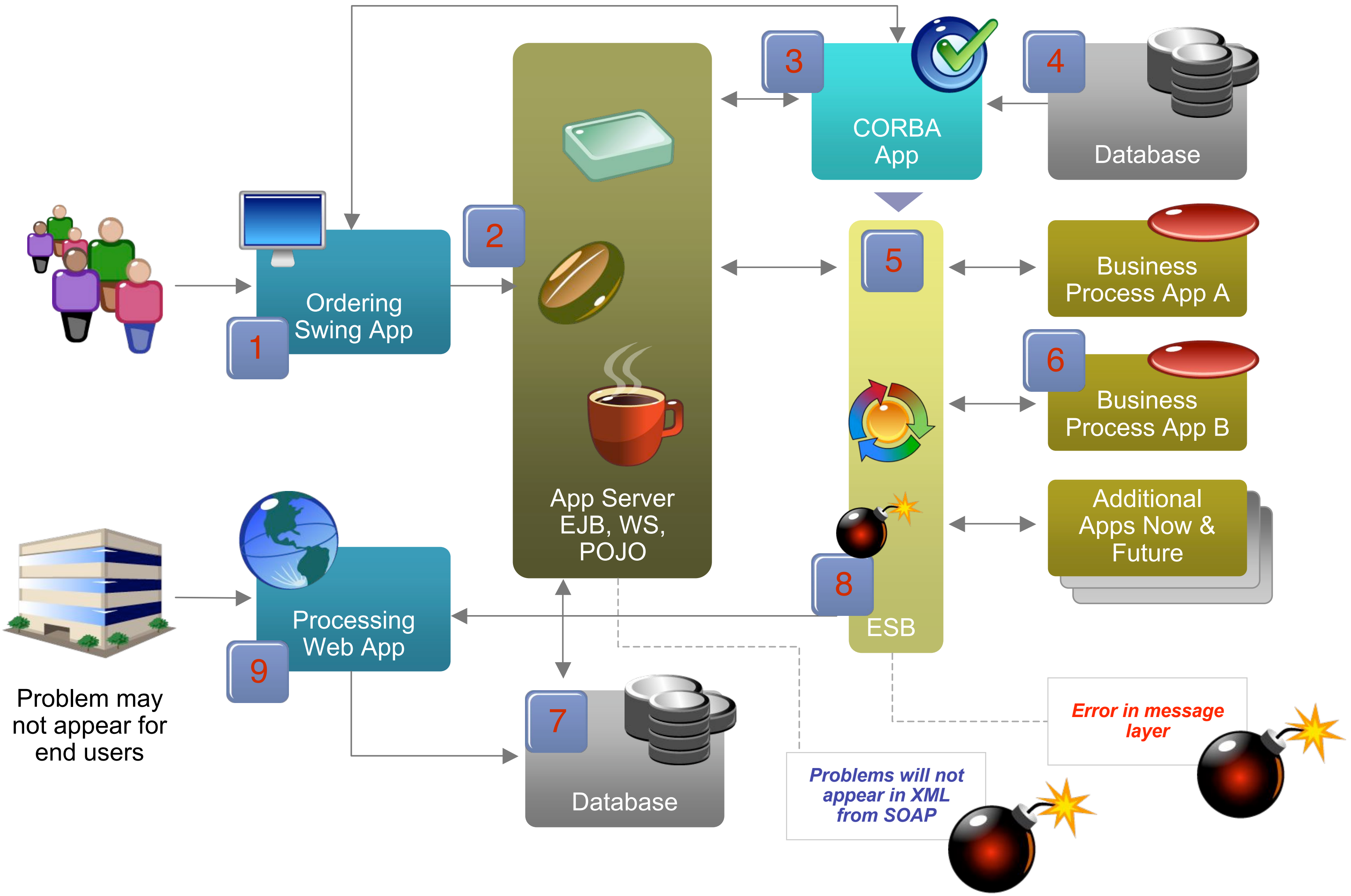
## Continuously Test

- Regress on existing functionality
- Add testing of new services to existing testing workflows

# Complete SOA Test Map - Simplified



# Complete Process Testing

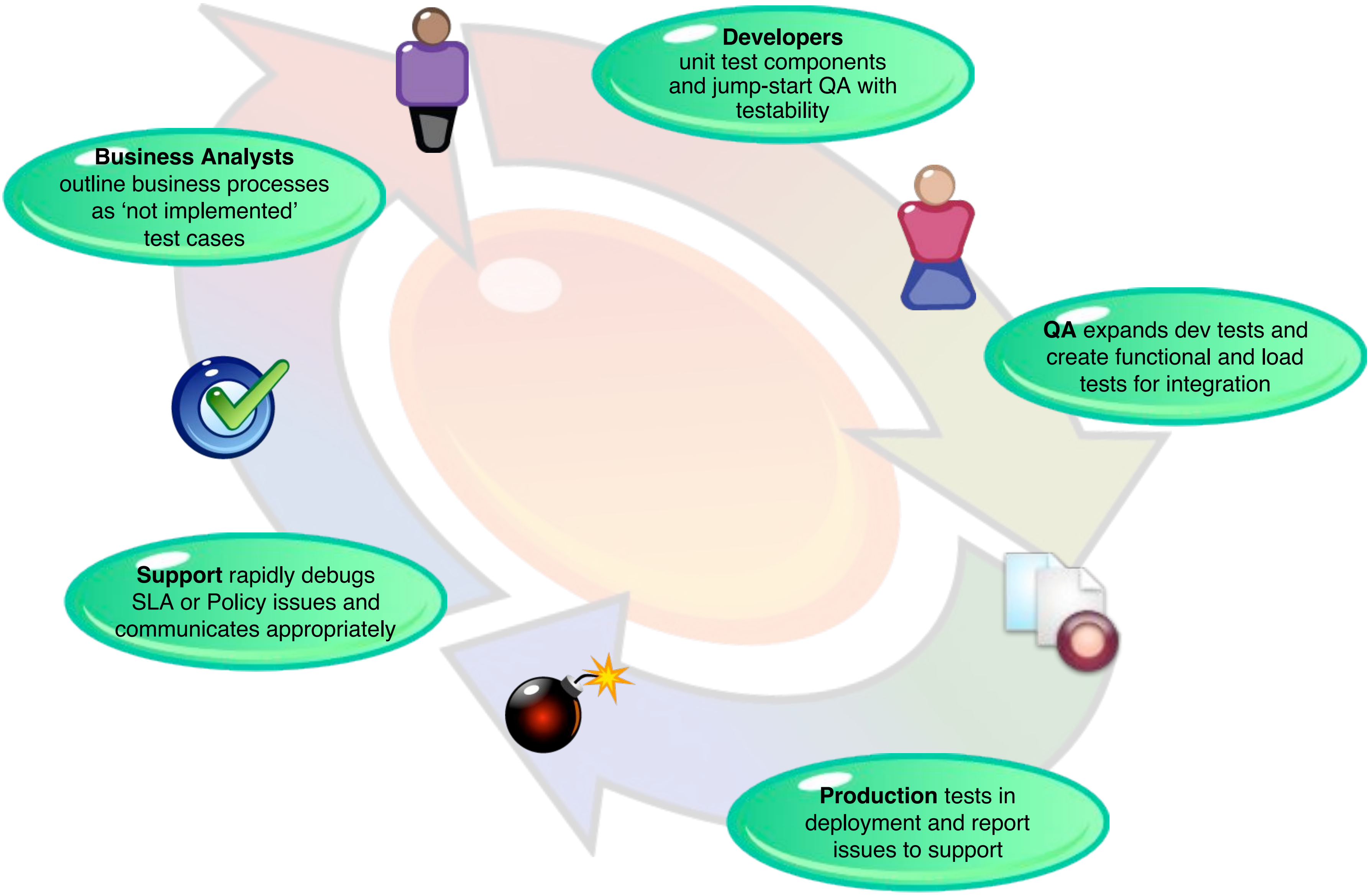


Problem may not appear for end users

*Problems will not appear in XML from SOAP*

*Error in message layer*

# Collaborative: SOA Requires Agile



**Business Analysts** outline business processes as ‘not implemented’ test cases

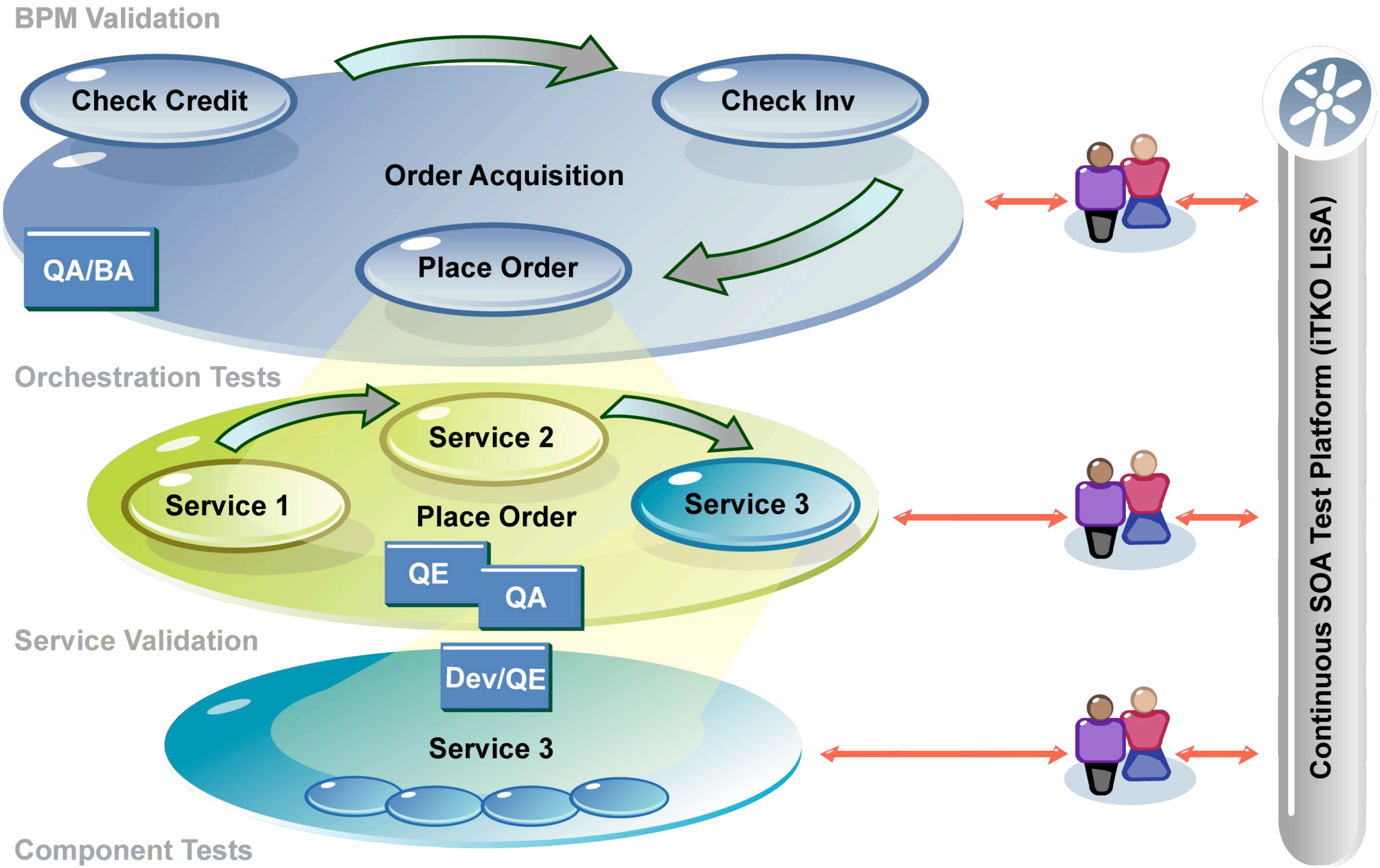
**Developers** unit test components and jump-start QA with testability

**QA** expands dev tests and create functional and load tests for integration

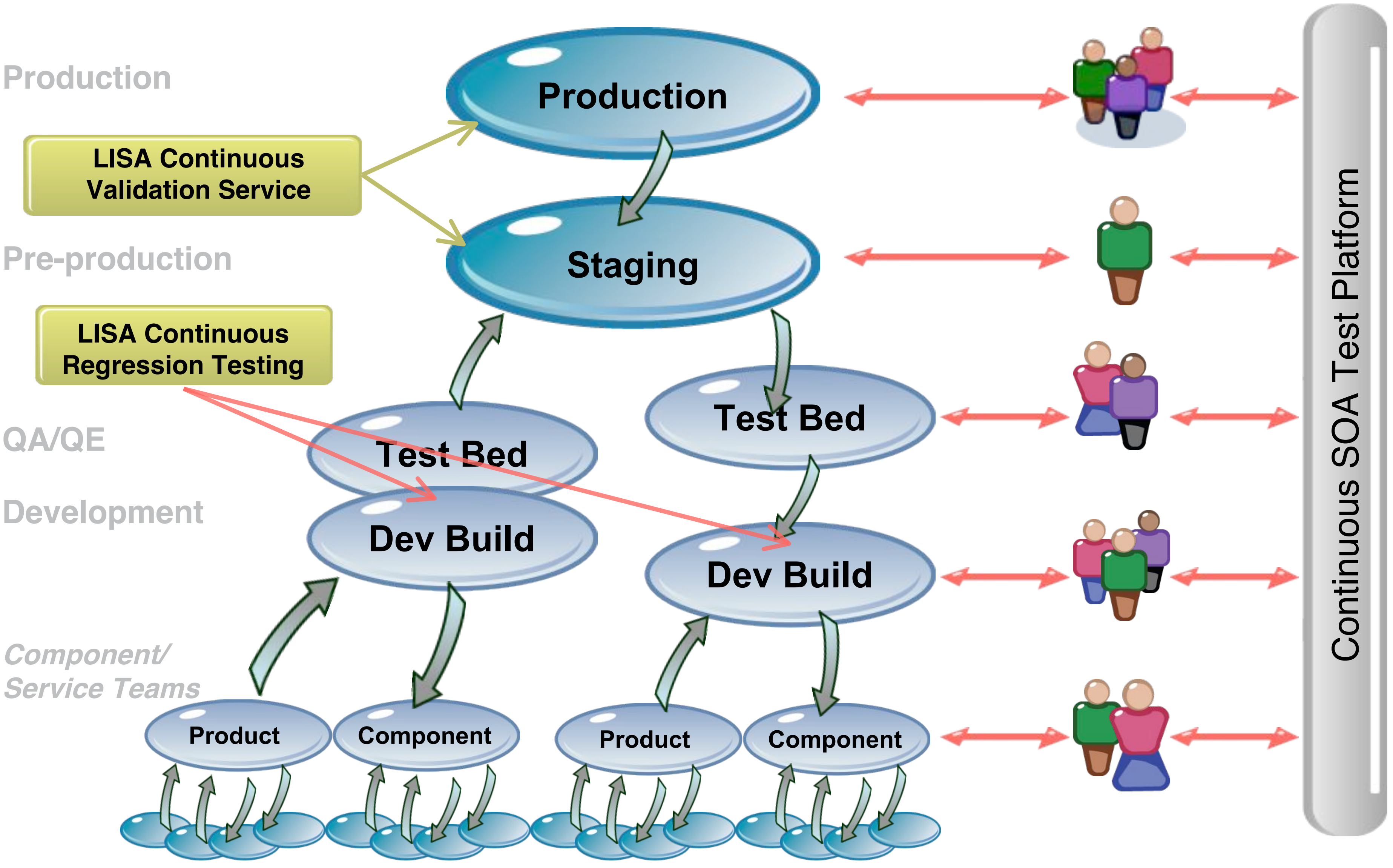
**Production** tests in deployment and report issues to support

**Support** rapidly debugs SLA or Policy issues and communicates appropriately

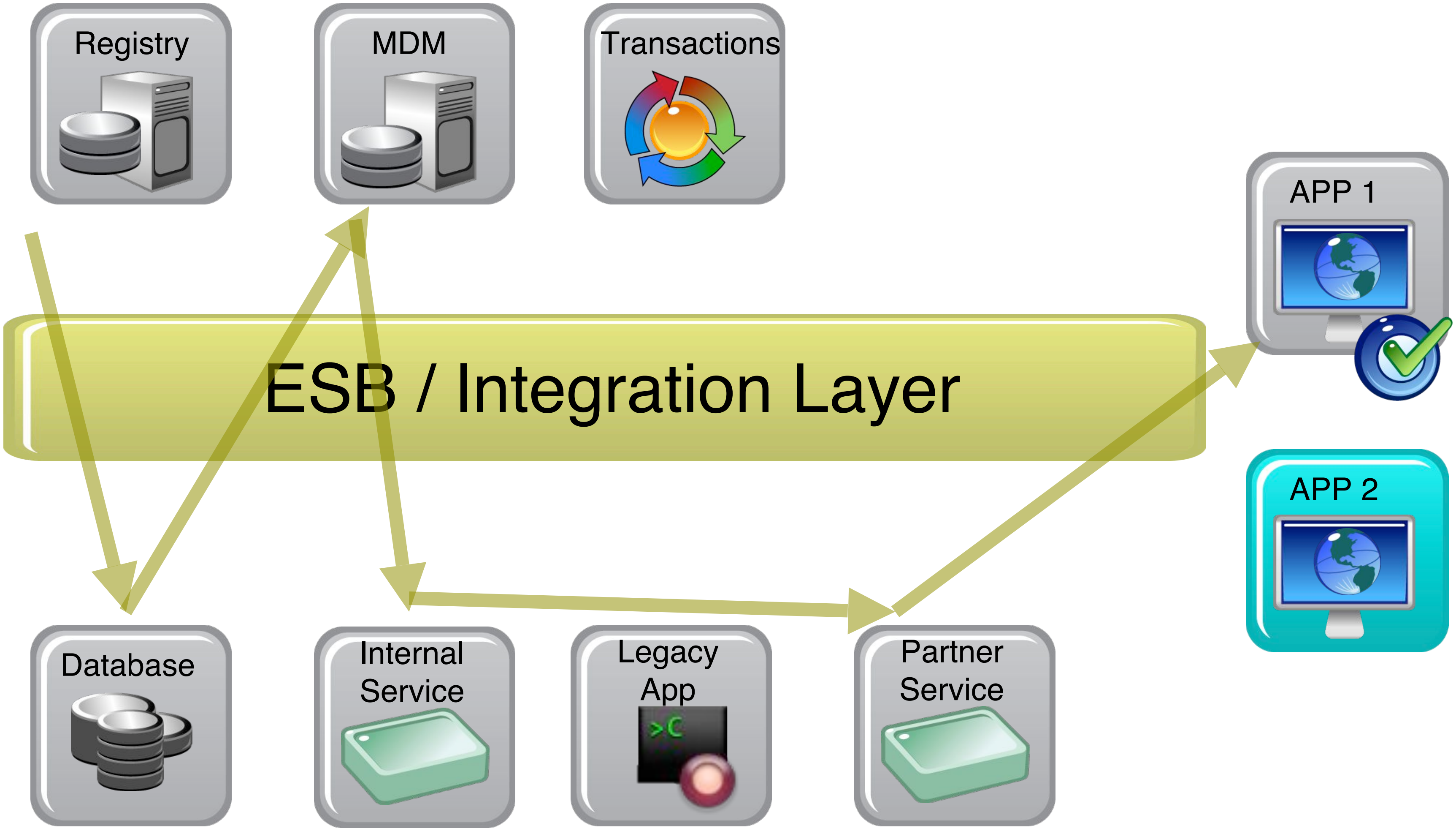
# Collaboration at every workflow level



# Continuous Integration = Continuous Testing

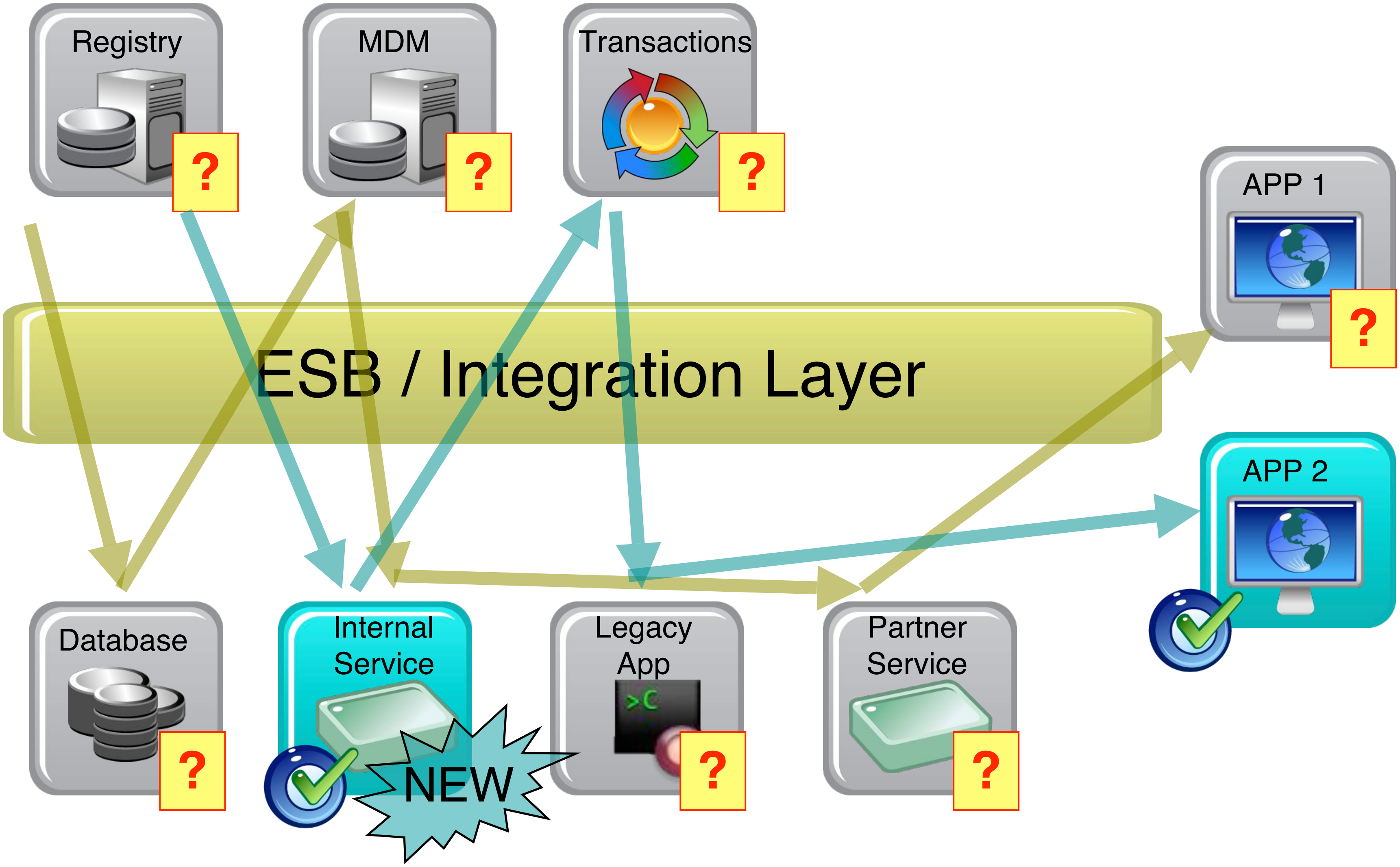


# Continuous: Unintended Consequences

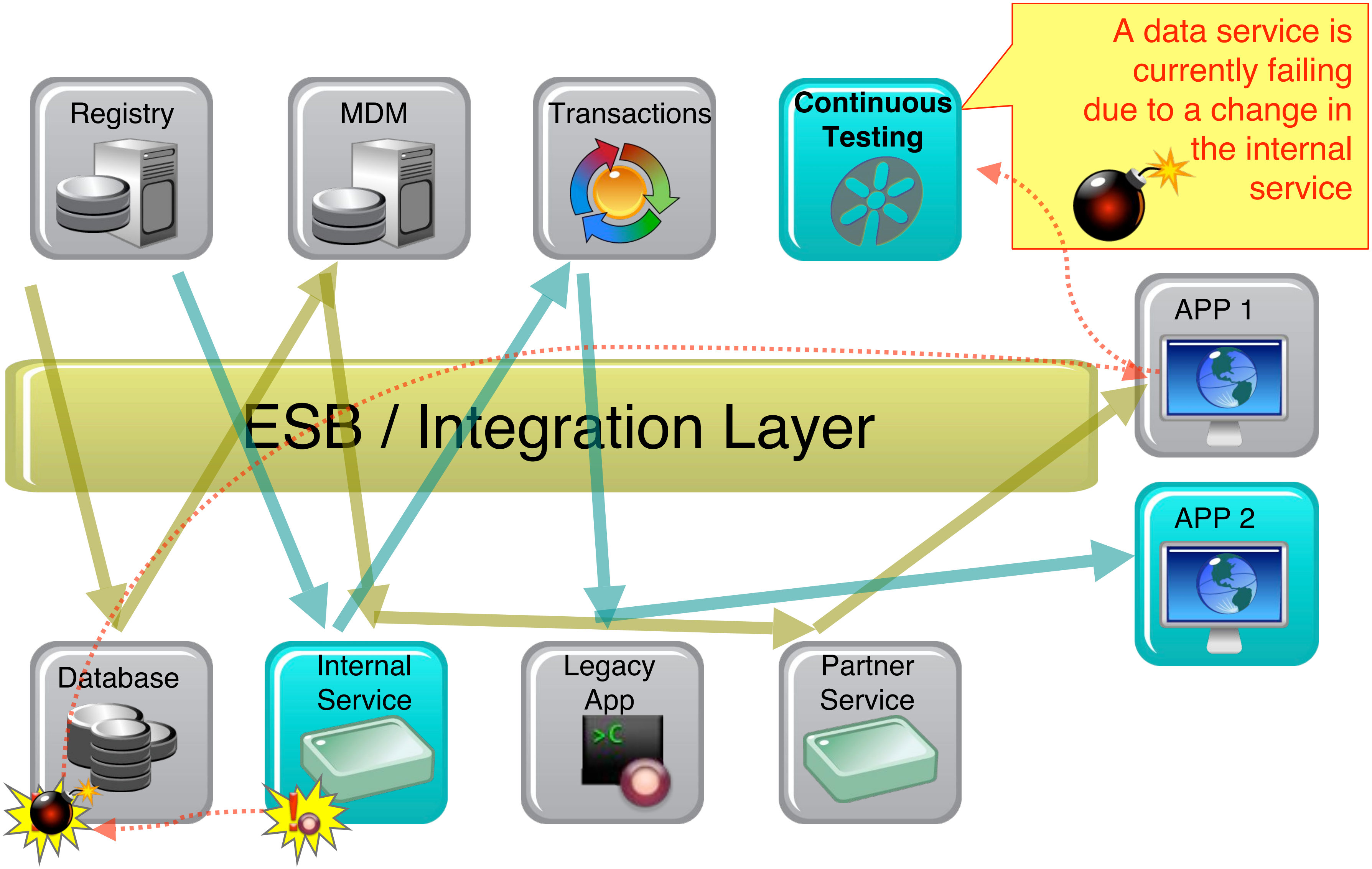


iTKO LISA™

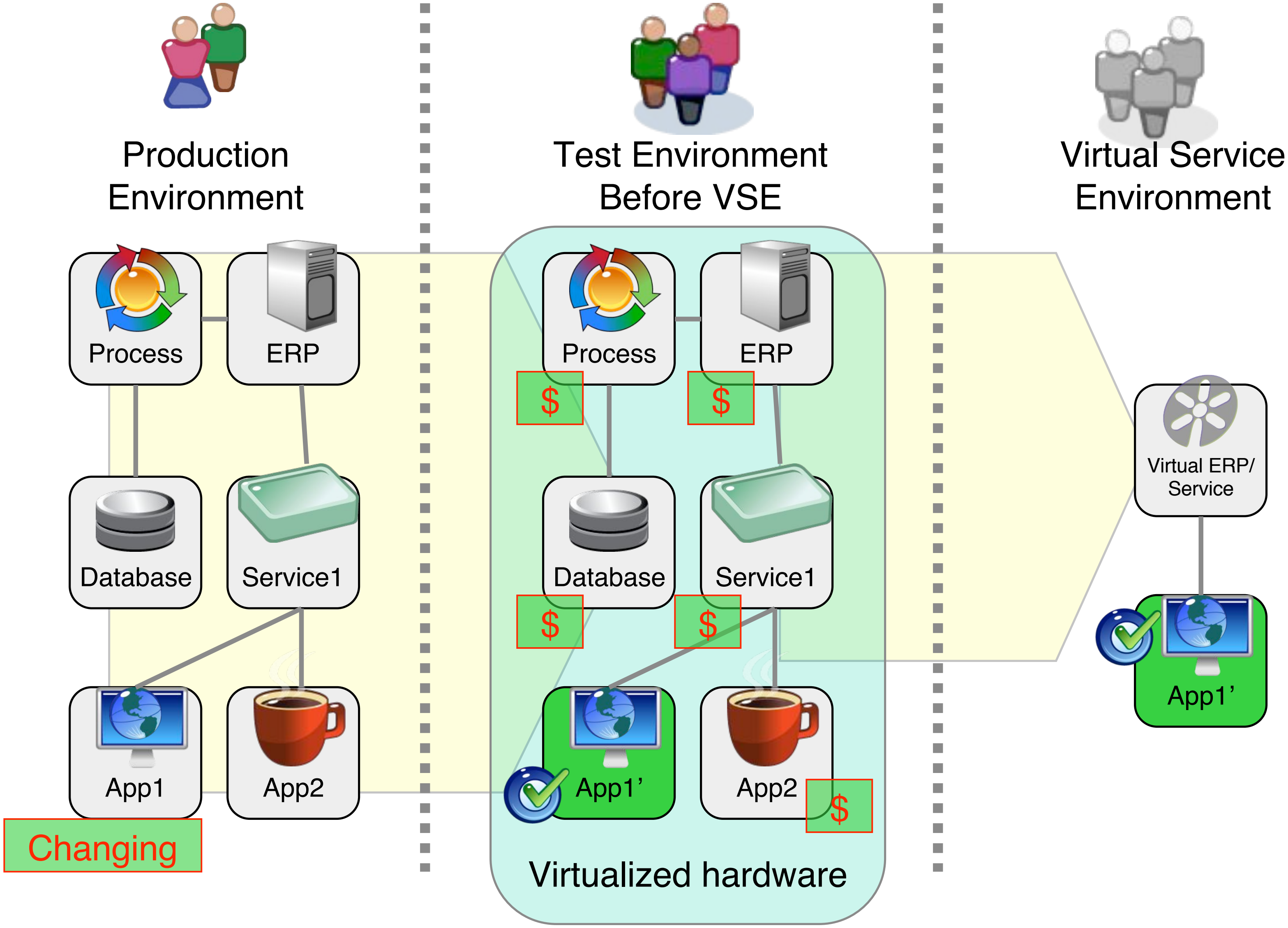
# Continuous: Unintended Consequences



# Continuous: Unintended Consequences



# Collaborative: Virtual Service Environment



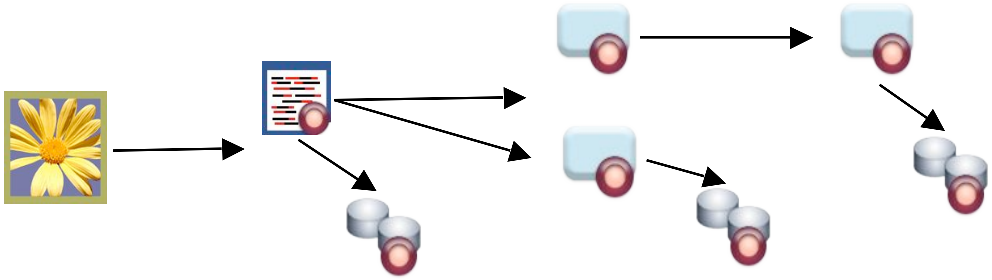
# LISA Virtual Service Environment: Example 1

- **Challenge: Agile Development with SOA**
  - During development, developers and QA teams must try to test against Services that are also currently under development
  - If a deployed version of a connected service is unavailable, QA must wait to test, or developers must code “mock” services to simulate these interactions.
  - Anytime the service that is being simulated changes, the test code needs to get updated. Testing and development therefore stall, reducing agility.
- **Solution**
  - LISA VSE automatically generates a Virtual Service by emulating the behaviors of the service, and its underlying implementation and data.
  - The Virtual Service is used by developers/QA to test their own services.
- **Value**
  - Enables Service Consumers to detect potential interaction problems early
  - Enables Service Producers to understand expected usages of their services even before they have delivered the actual service
  - Allows highly parallel, agile development and testing collaboration across organization, saving significant cost/time and averting risk earlier.

# LISA Virtual Service Environment: Example 2

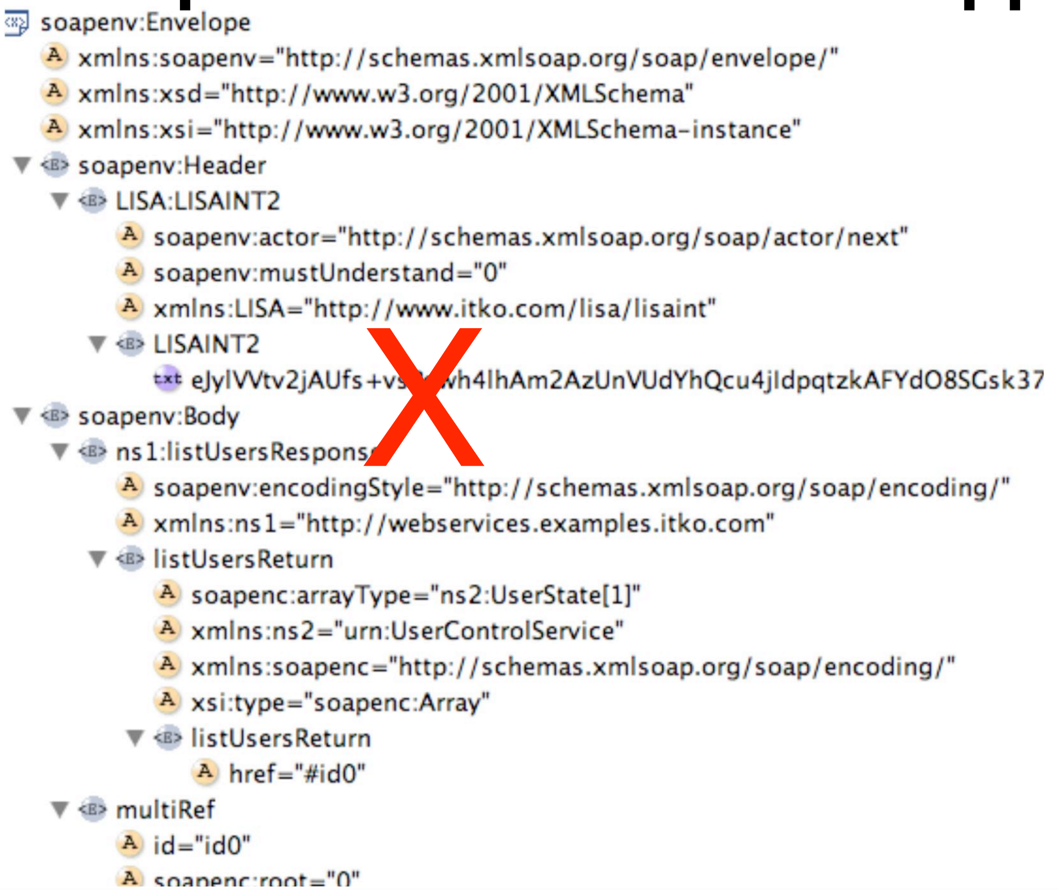
- **Challenge: Testing Deployed Integrations**
  - During development and integration test cycles, teams must rely on complex application components that are in production for complete workflow testing, but these are very difficult/expensive to replicate
  - These applications (Mainframes, services, databases, etc.) can become a major constraint, or become unavailable if several different teams try and test their code against them
  - Also, even if the applications are replicated on Virtualized hardware, each new test bed is extremely costly to license, install, support and maintain.
- **Solution**
  - Developers and Testers get a Virtual Service that emulates most of the behaviors of the entire multi-tier service-based application, eliminating the need to replicate and maintain a new test environment to test the service under development
  - LISA VSE reduces the number of test beds and eliminates the need to test directly against the constrained application by simulating connected services and applications through an intelligent record and replay mechanism
- **Value**
  - Customers save millions of dollars in hardware, software and maintenance cost, without compromising quality

# Visibility (LISA Pathfinder)

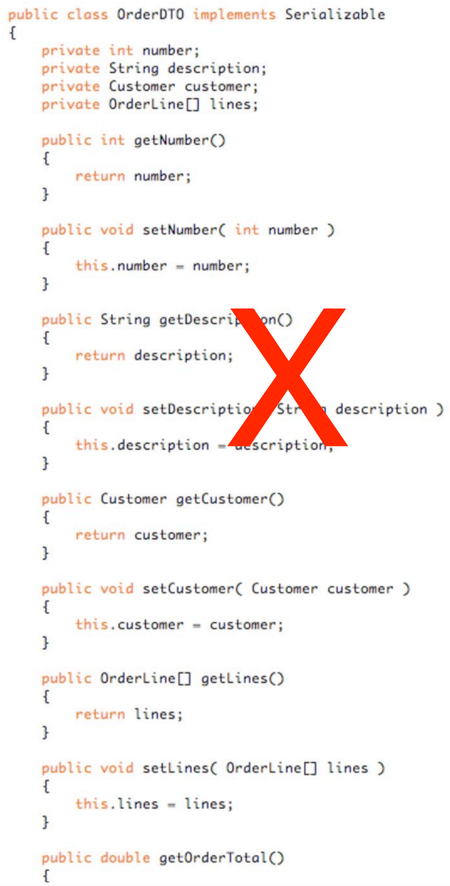


- Customer Challenge
  - QA personnel, used to testing only user interfaces, are being asked to test headless services and applications
  - In most cases, they have very little idea, if any, of what to test
    - they cannot see what they need to test
- Solution
  - LISA Pathfinder enables them to visualize all the components that are invoked by any transaction
  - It further enables them to generate test case shells automatically that can be edited and completed within the LISA Workstation
- Value
  - Pathfinder can significantly reduce the time to create complex test cases
  - Less technical QA personnel will be **able to test invisible services** and components with relative ease
  - **Rapid Identification of root causes of failures**

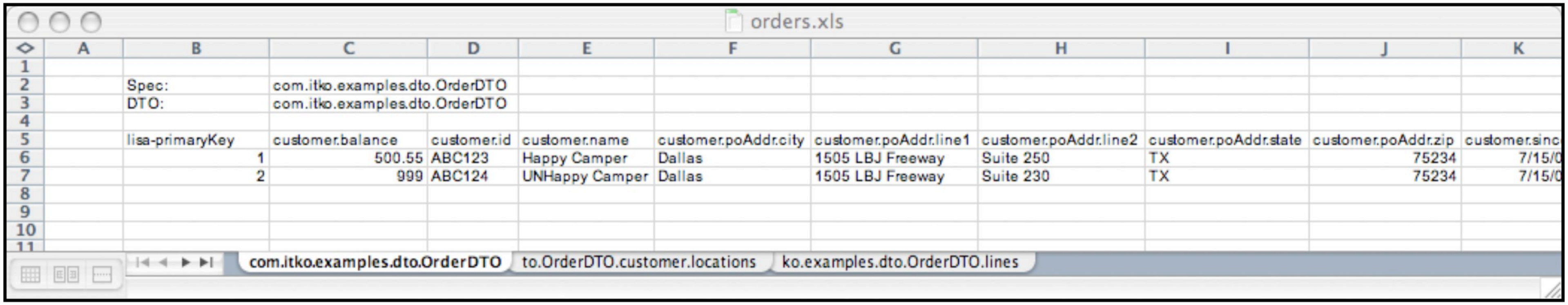
# Complex Data Scenario Support



```
soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  ▼ soapenv:Header
    ▼ LISA:LISAINT2
        soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
        soapenv:mustUnderstand="0"
        xmlns:LISA="http://www.itko.com/lisa/lisaint"
      ▼ LISAINT2
          txt eJylVVtv2jAUfs+vs[illegible]wh4lhAm2AzUnVUdYhQcu4jIdpqtzkAFYdO8SGsk37
  ▼ soapenv:Body
    ▼ ns1:listUsersResponse
        soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:ns1="http://webservices.examples.itko.com"
      ▼ listUsersReturn
          soapenc:arrayType="ns2:UserState[1]"
          xmlns:ns2="urn:UserControlService"
          xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
          xsi:type="soapenc:Array"
        ▼ listUsersReturn
            href="#id0"
    ▼ multiRef
        id="id0"
        soapenc:root="0"
```



```java
public class OrderDTO implements Serializable
{
    private int number;
    private String description;
    private Customer customer;
    private OrderLine[] lines;

    public int getNumber()
    {
        return number;
    }

    public void setNumber( int number )
    {
        this.number = number;
    }

    public String getDescription()
    {
        return description;
    }

    public void setDescription( String description )
    {
        this.description = description;
    }

    public Customer getCustomer()
    {
        return customer;
    }

    public void setCustomer( Customer customer )
    {
        this.customer = customer;
    }

    public OrderLine[] getLines()
    {
        return lines;
    }

    public void setLines( OrderLine[] lines )
    {
        this.lines = lines;
    }

    public double getOrderTotal()
    {
```



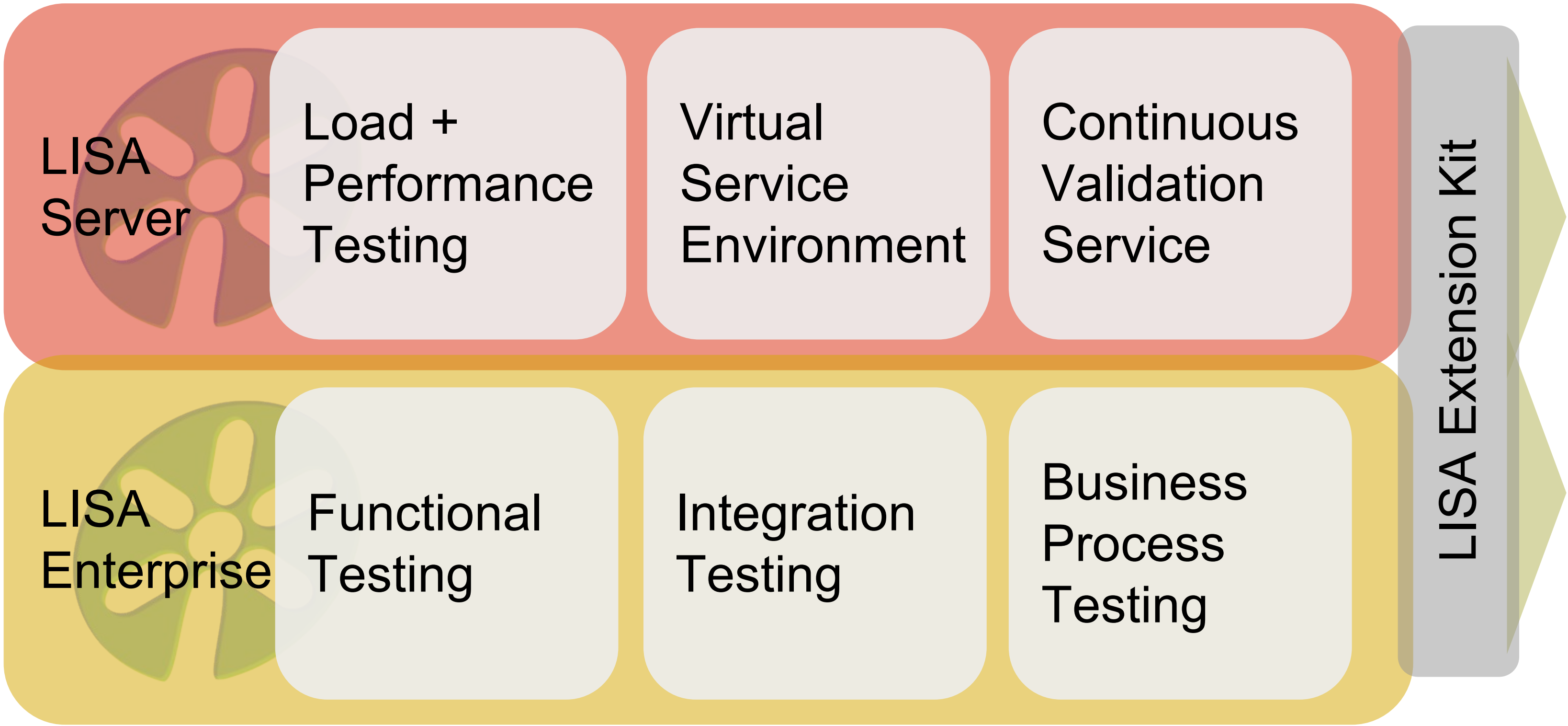
orders.xls

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | |
| 2 | | Spec: | com.itko.examples.dto.OrderDTO | | | | | | | | |
| 3 | | DTO: | com.itko.examples.dto.OrderDTO | | | | | | | | |
| 4 | | | | | | | | | | | |
| 5 | | lisa-primaryKey | customer.balance | customer.id | customer.name | customer.poAddr.city | customer.poAddr.line1 | customer.poAddr.line2 | customer.poAddr.state | customer.poAddr.zip | customer.sinc |
| 6 | | 1 | 500.55 | ABC123 | Happy Camper | Dallas | 1505 LBJ Freeway | Suite 250 | TX | 75234 | 7/15/0 |
| 7 | | 2 | 999 | ABC124 | UNHappy Camper | Dallas | 1505 LBJ Freeway | Suite 230 | TX | 75234 | 7/15/0 |
| 8 | | | | | | | | | | | |
| 9 | | | | | | | | | | | |
| 10 | | | | | | | | | | | |
| 11 | | | | | | | | | | | |

com.itko.examples.dto.OrderDTO
to.OrderDTO.customer.locations
ko.examples.dto.OrderDTO.lines

# LISA 4 Product Suite

## iTKO LISA SOA Testing Suite



LISA Server

- Load + Performance Testing
- Virtual Service Environment
- Continuous Validation Service

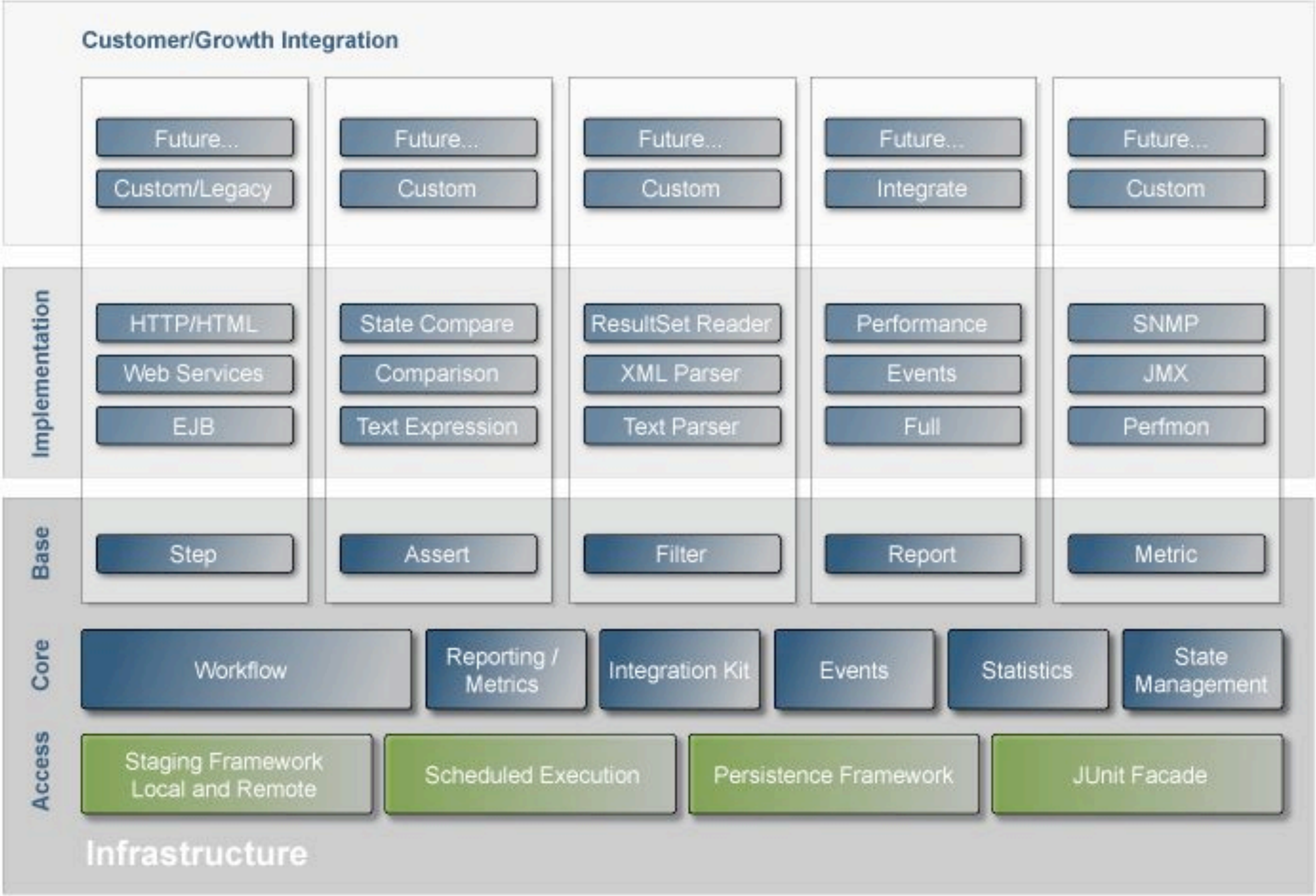
LISA Enterprise

- Functional Testing
- Integration Testing
- Business Process Testing

LISA Extension Kit

# LISA’s Test Extensibility Framework

iTKO LISA™



Customer/Growth Integration

- Future... / Custom/Legacy
- Future... / Custom
- Future... / Custom
- Future... / Integrate
- Future... / Custom

Implementation

- HTTP/HTML / Web Services / EJB
- State Compare / Comparison / Text Expression
- ResultSet Reader / XML Parser / Text Parser
- Performance / Events / Full
- SNMP / JMX / Perfmon

Base

- Step
- Assert
- Filter
- Report
- Metric

Core

- Workflow
- Reporting / Metrics
- Integration Kit
- Events
- Statistics
- State Management

Access

- Staging Framework Local and Remote
- Scheduled Execution
- Persistence Framework
- JUnit Facade

Infrastructure