Sun Systemic Security
Security Patterns for IT Architecture

Glenn Brunette
Distinguished Engineer
Sun Microsystems, Inc.

Agenda

- Architectural Patterns
- Sun Systemic Security
- Security Principles
- Security Building Blocks and Patterns
- References

Special thanks to Jason Carolan, Mikael Lofstrand, and John Stanford for their great work on patterns and Joel Weise and Rafat Alvi for their contributions to the Sun Systemic Security program.

Architectural Patterns

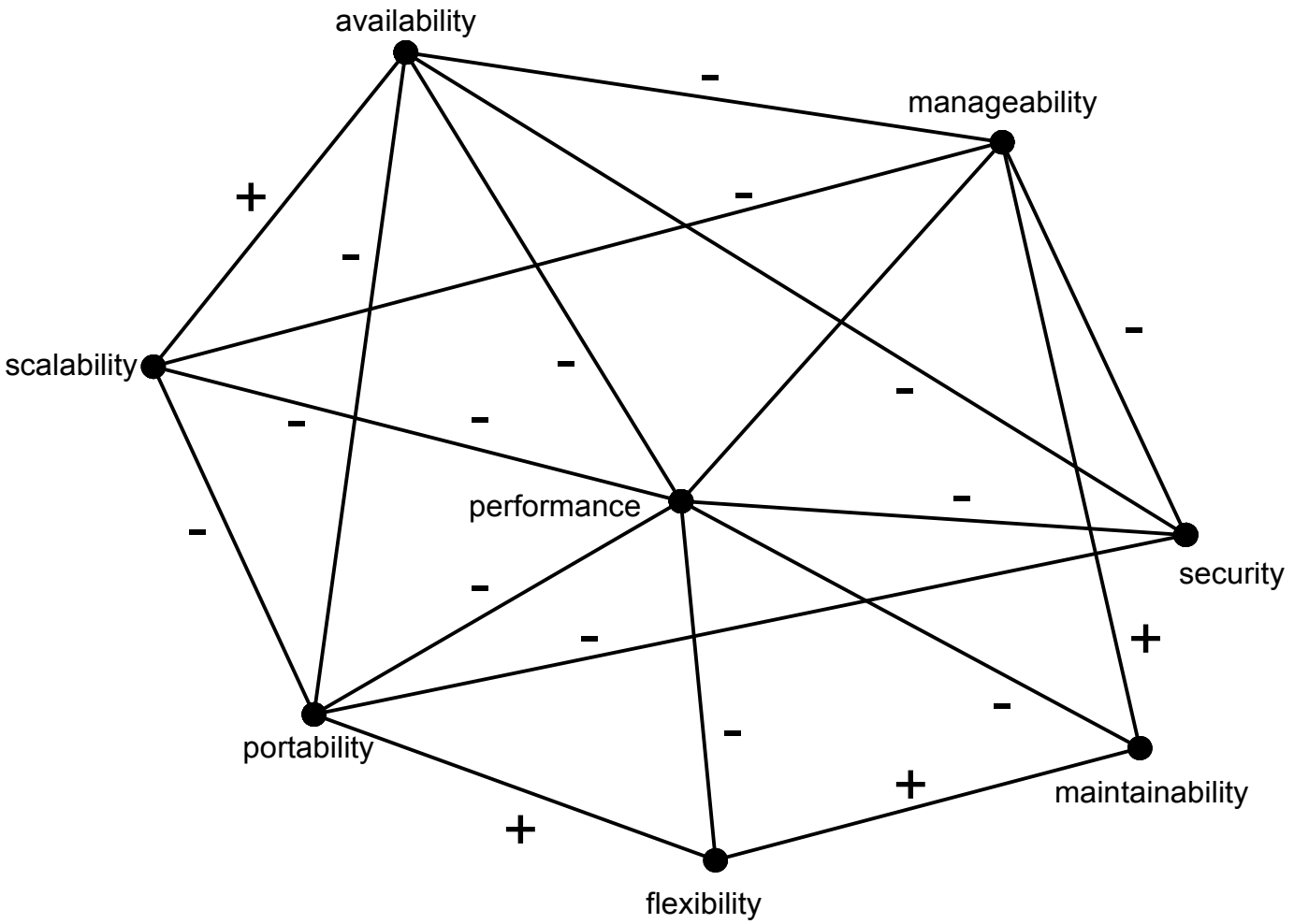
- What is an Architectural Pattern?

  The core of a solution to a repeated IT problem described in such a way that you can use the solution a million times over, without ever necessarily implementing the solution the same way twice (ref: Alexander)

- What does an Architectural Pattern describe?
  - Name - Unique, descriptive identifier
  - Problem - Design problem to be solved
  - Context - Environment of the pattern
  - Forces - Reasons and motivation for selection
  - Strategies - Describe different implementations
  - Solution - Describe approach to solving problem
  - Consequences - Describe benefits & trade-offs of the different approaches

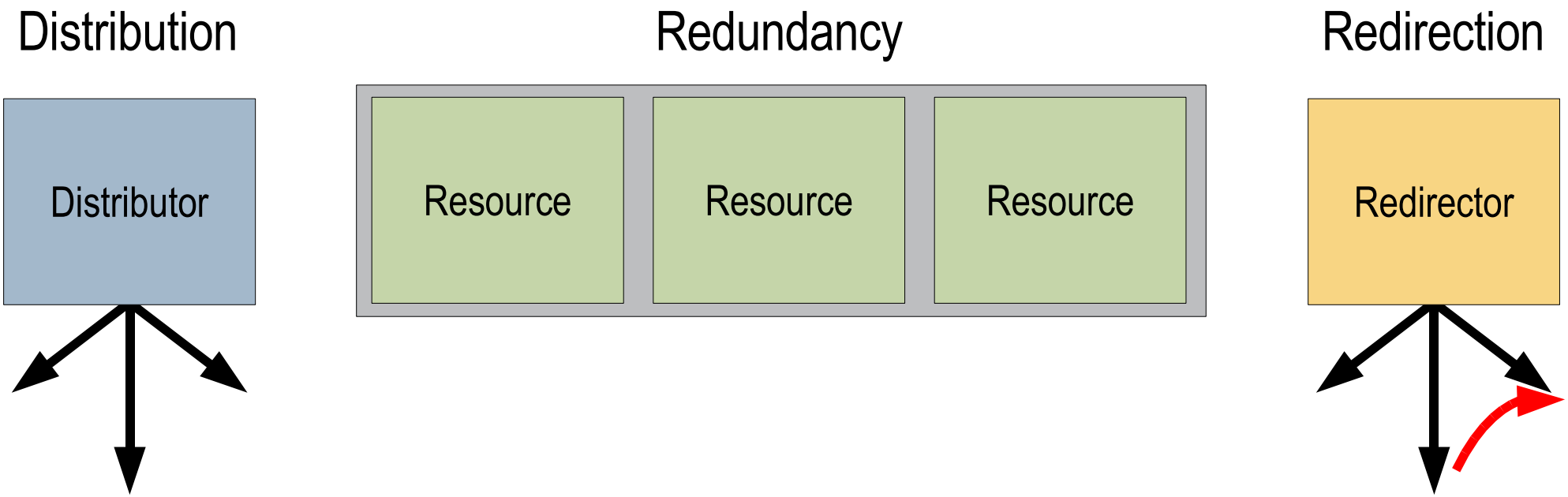
Example: Architectural Forces

Non-functional Characteristics Forces Interaction Model



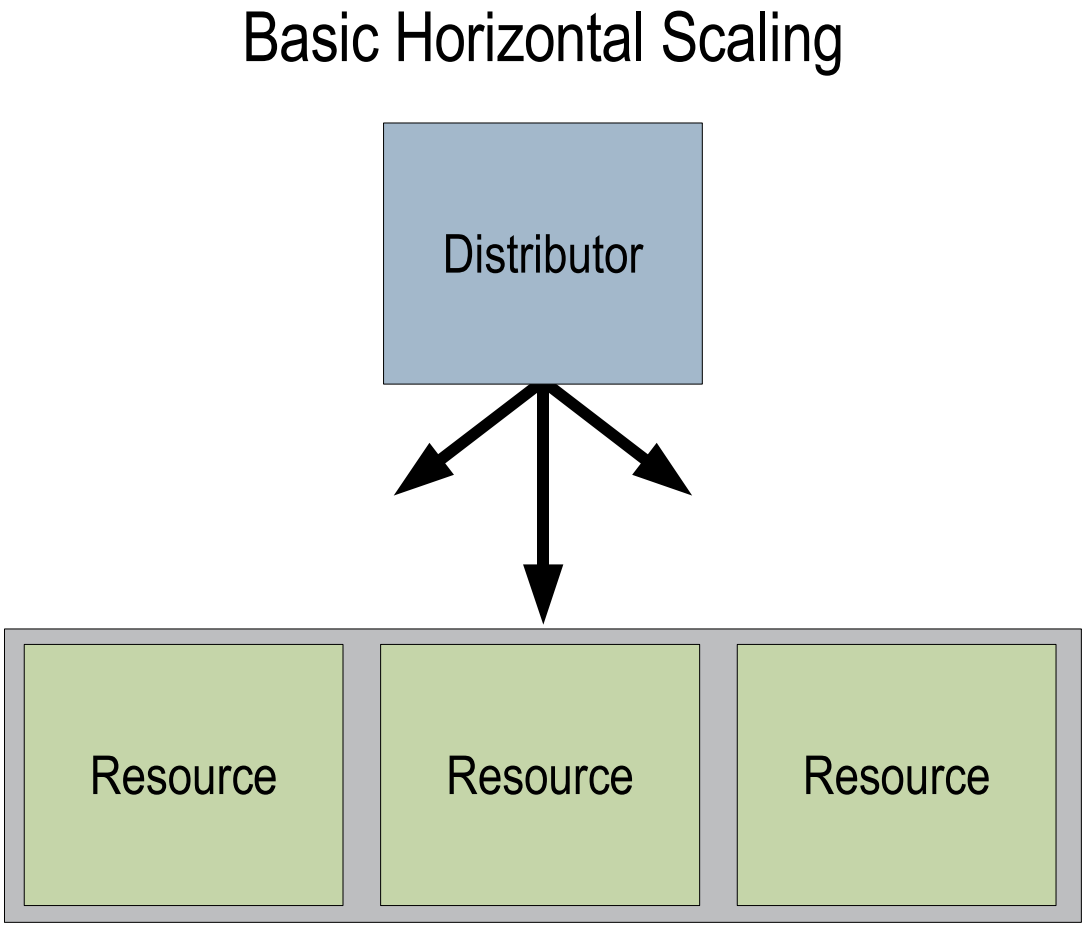
Reference: Architecting Enterprise Solutions: Patterns for High-Capability Internet-based Systems by Paul Dyson and Andrew Longshaw

Example: Architectural Patterns

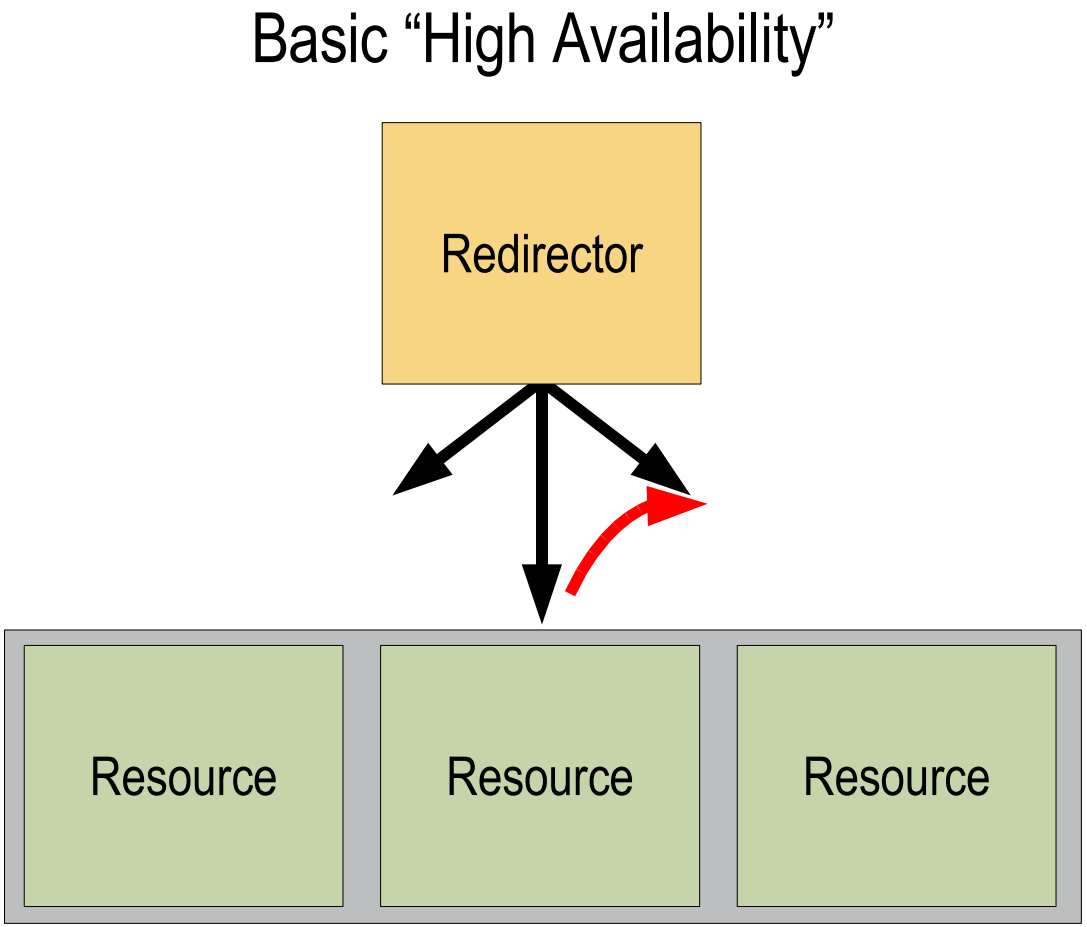


Principles applied to Problems and Context == Patterns

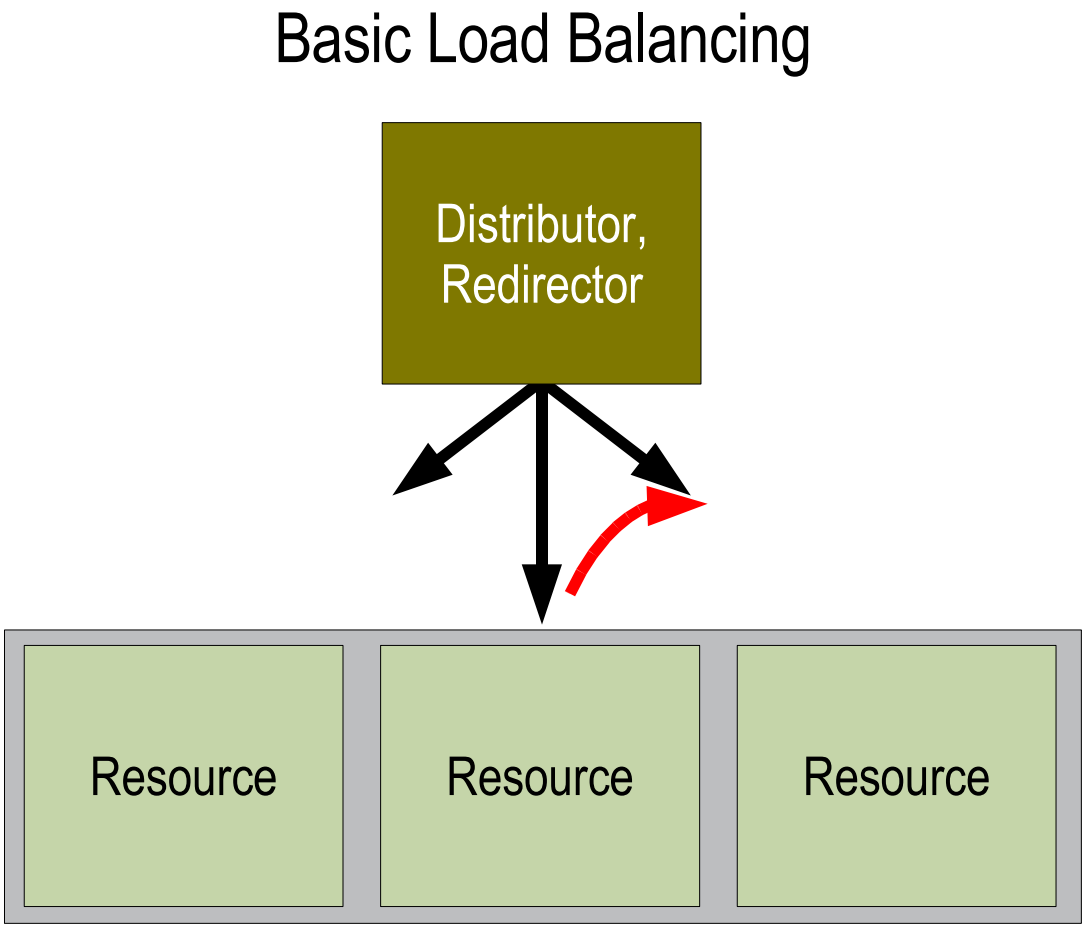
Example: Architectural Strategies



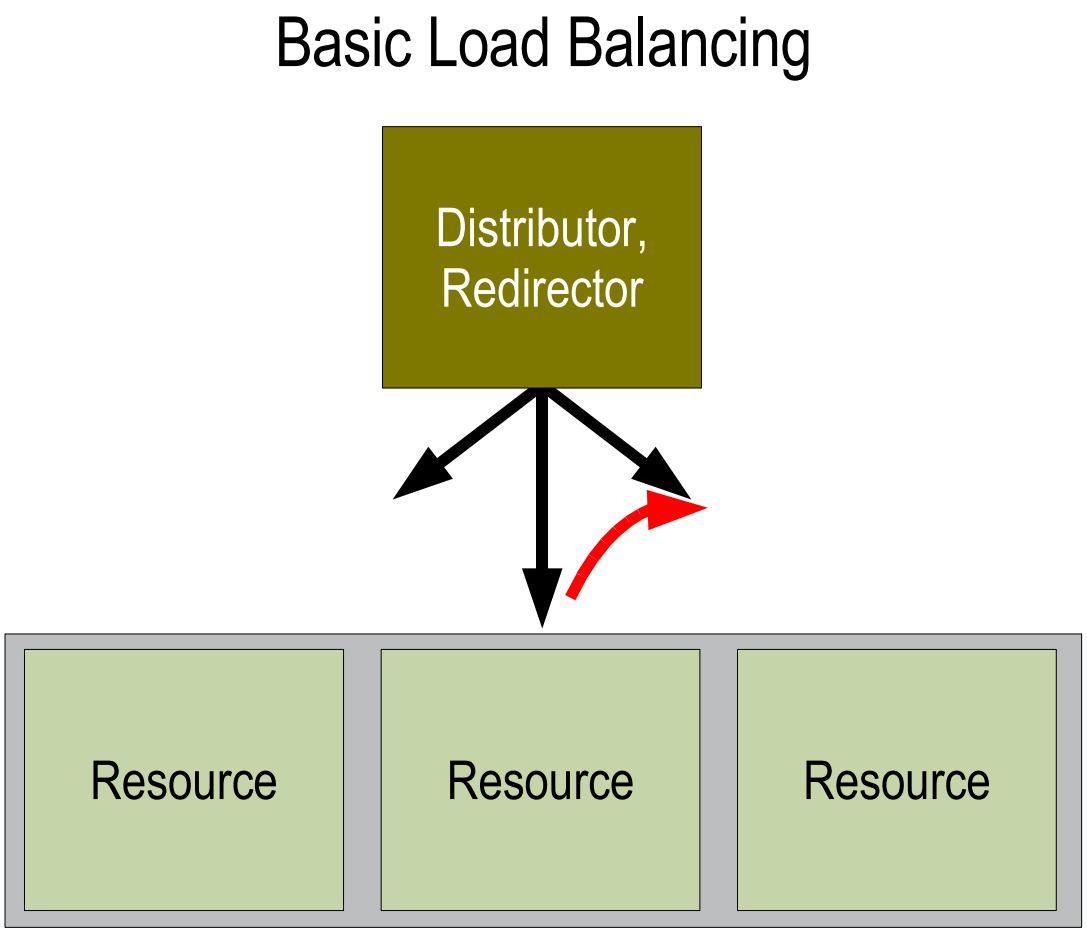
Example: Architectural Strategies



Example: Architectural Strategies

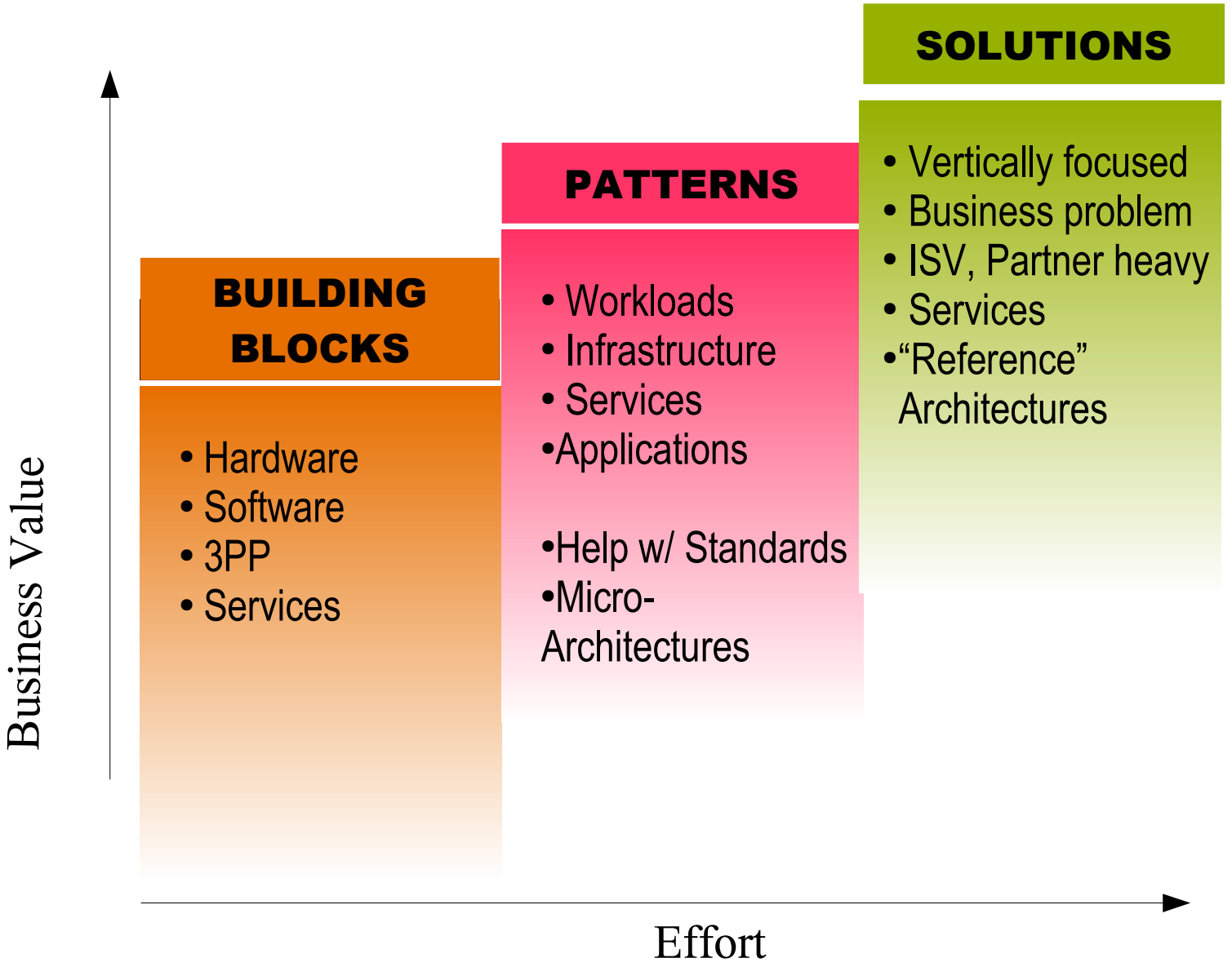


Example: Architectural Strategies



- "Solutions" derived from strategies will evolve over time:
  - RR-DNS
  - Resonate, Local Director
  - Alteon, Arrowpoint
  - Sun Secure App. Switch
- Context and forces will help guide the selection of possible strategies.
- Strategies can leverage individual building blocks, patterns and even other solutions.

From Building Blocks to Solutions

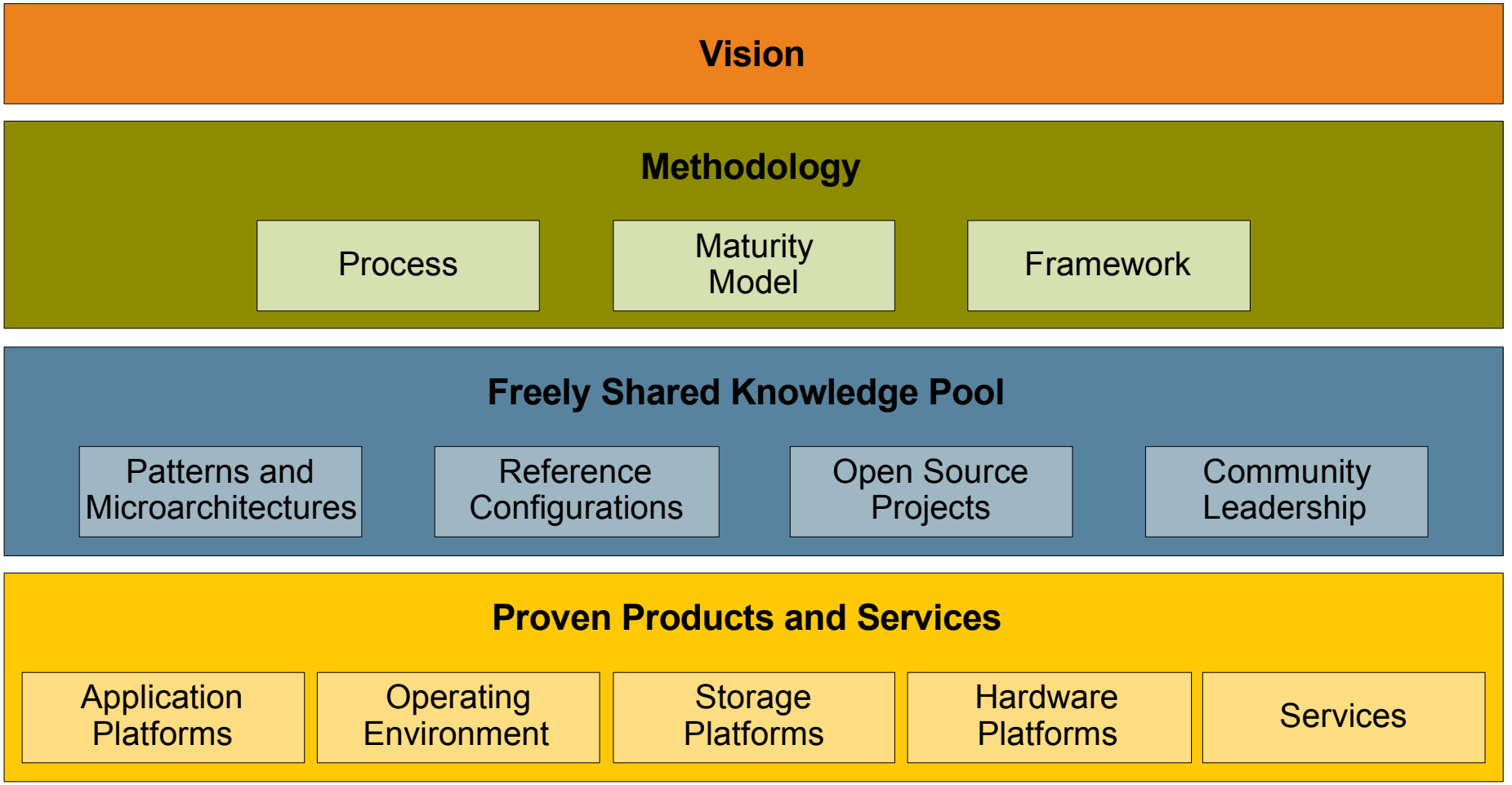


Pattern Fallacies

- Patterns are applicable only to up-front design in greenfield projects.
- Patterns are focused at a fixed-level of abstraction.
- Patterns are not useful for practical design.
- Patterns are limited to specific domains.
- Patterns can be practiced and adopted in individual isolation.
- Patterns are a “solution to a problem”.
- Patterns are concrete, reusable design ideas.
- Patterns eliminate need for creative thinking.

So, what does this have to do with security?

Sun Systemic Security



Architectural Approach to Security

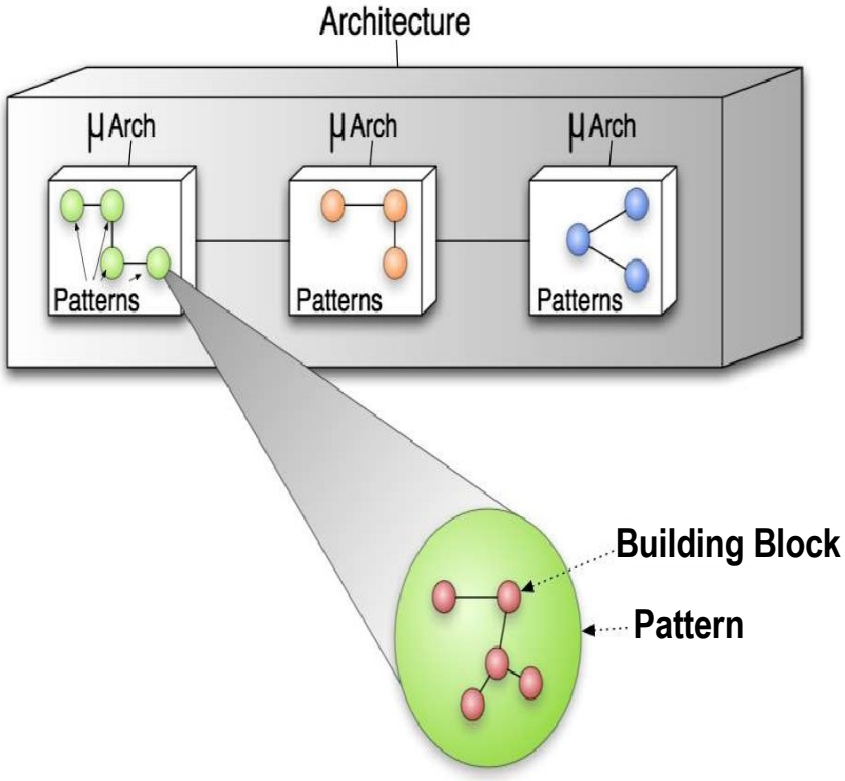
- Secure and Compliant
  - > Achieves balance between contexts and forces with a special focus on security, privacy, and regulatory compliance
  - > Applies time-tested security principles to new/existing architectural approaches
  - > Embraces the holistic and systemic nature of security - policy, process, education, architecture and technology, etc.
- Repeatable and Sustainable
  - > Facilitates a structured approach to consolidation, standardization, automation and optimization and thereby continuous improvement
  - > Amplifies the value obtained through IT governance and ITSM
- Transformational
  - > Improves upon organizational security maturity levels
  - > Delivers small, modular “building blocks” to facilitate iterative refinement

Architectural Security Principles

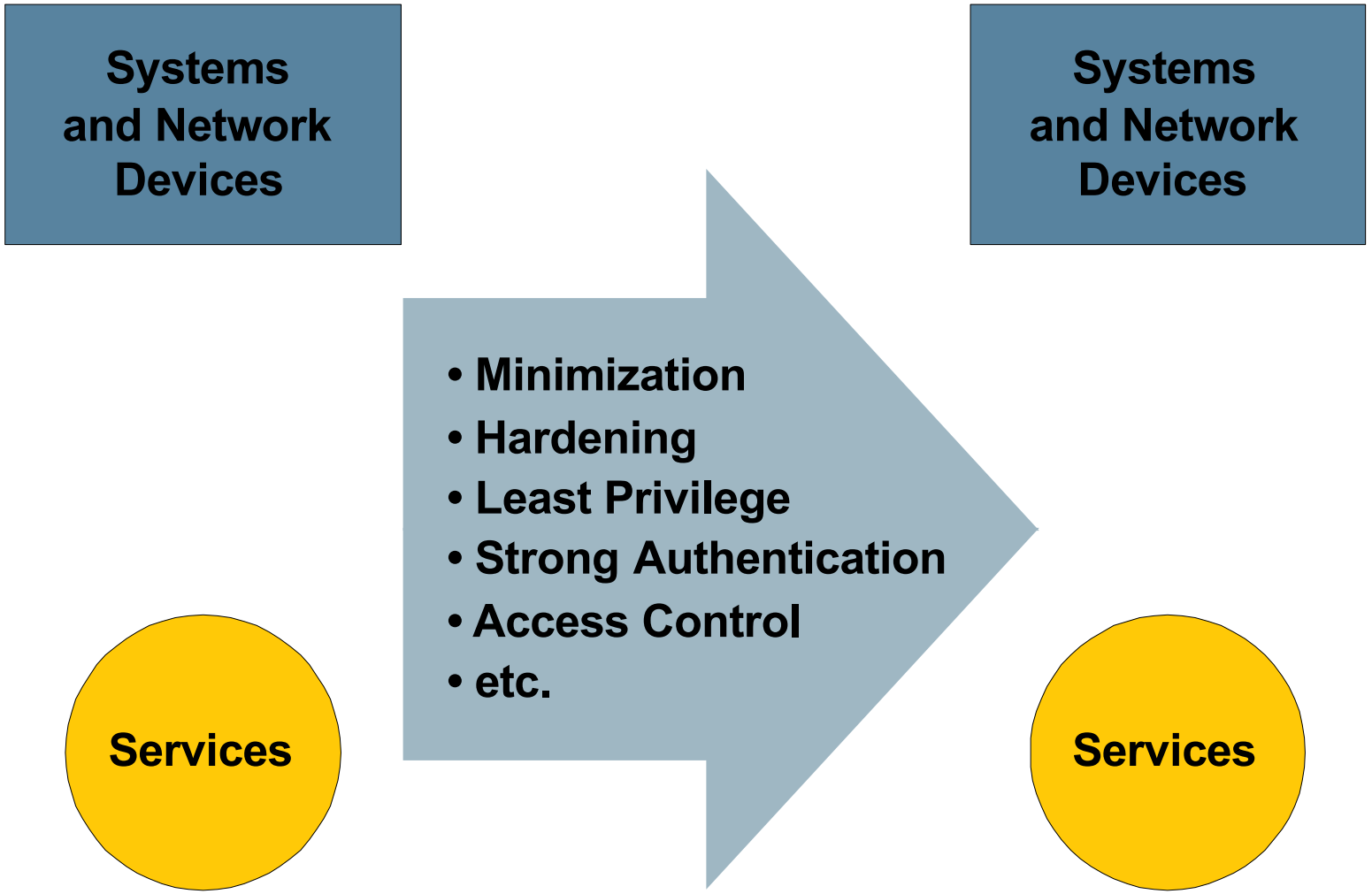
- Starts with the typical general (meta) principles of:
  - Confidentiality
  - Integrity
  - Availability
  - Accountability
  - Non-repudiation
- also maintain a focus on architectural principles:
  - Self-Preservation
  - Defense in Depth
  - Least Privilege
  - Compartmentalization
  - Proportionality
  - Interface Driven

Security Building Blocks and Patterns

- Secure Components
- Secure Execution Containers
- Secure Network Enclaves
- Shared Infrastructure Services
- Shared Application Services
- Secure Presentation Services
- Secure Desktop Services

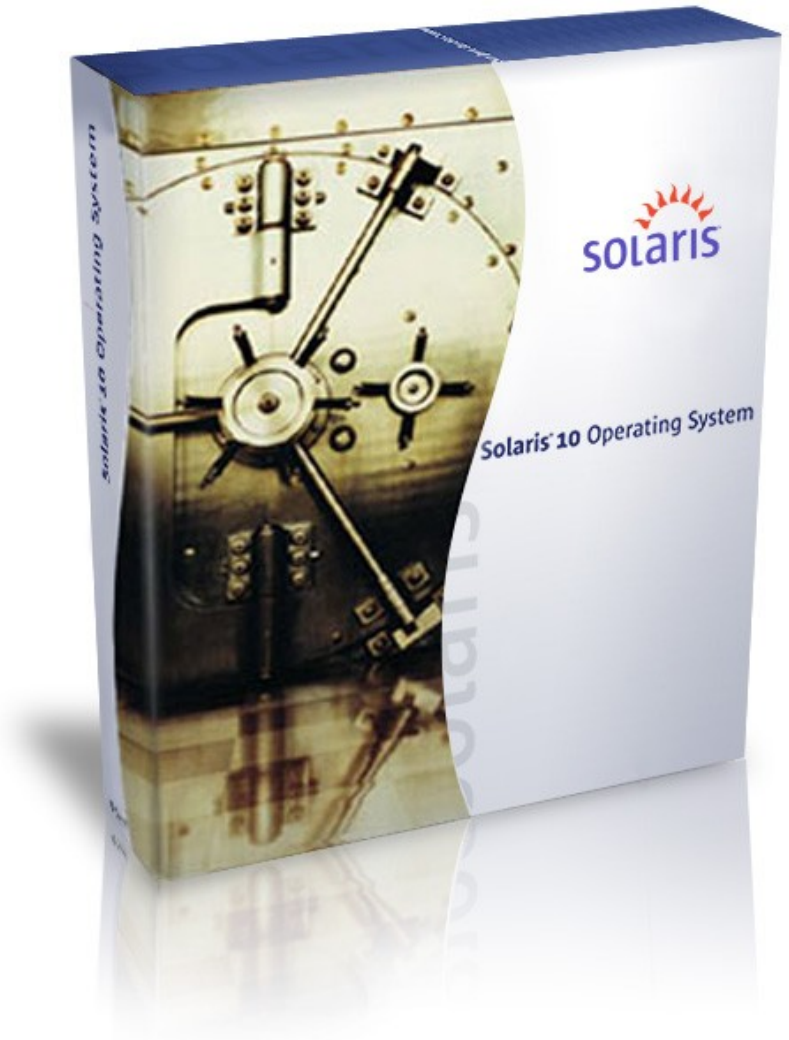


Secure Components



Secure Components

Example: Solaris 10 Operating System



- Secure Execution
- User Rights Management
- Process Rights Management
- Solaris Containers
- Cryptographic Framework
- IP Filter
- Solaris Trusted Extensions

Secure Components

Example: Solaris Security Toolkit (JASS)

- Solaris 10 Support
  - Integration with the Service Management Facility (SMF)
  - Integration with Solaris Containers (Zones)
  - New Access Control Settings (TCP Wrappers, IP Filter, etc.)
  - New Password Format and Composition Settings
- Compatible with SPARC, x86/x64
- Fully supported (as of version 4.1)
- Industry-based Settings and Recommendations

Secure Execution Containers

- Provision Services into Secure Execution Containers
  - Solaris, J2EE, Hardware (Crypto) Containers, etc.

- Required degree of assurance dictates level of separation:
  - Physical (e.g., Separate Systems)
  - Electrical (e.g., Domains)
  - Logical separation (e.g., Solaris Containers)

Secure Execution Containers

Example: Solaris 10 Zones

- Restricted Operations for Enhanced Security
  - Access raw memory, DTrace, promiscuous mode snooping, altering network interface and route information, manipulating devices and kernel modules, altering system time, etc.
- Enforcement with Assurance
  - Spare root zones, IP Filter, restricted mounts, etc.
- Resource Control and Management
  - CPU, disk, networking, etc.
- Observability with Integrity
  - BART, Solaris Auditing, etc.

... plus all the normal benefits of Solaris ...

Composite Security Strategy Example

Secure Component

Composite Security Strategy Example

Secure Execution Container

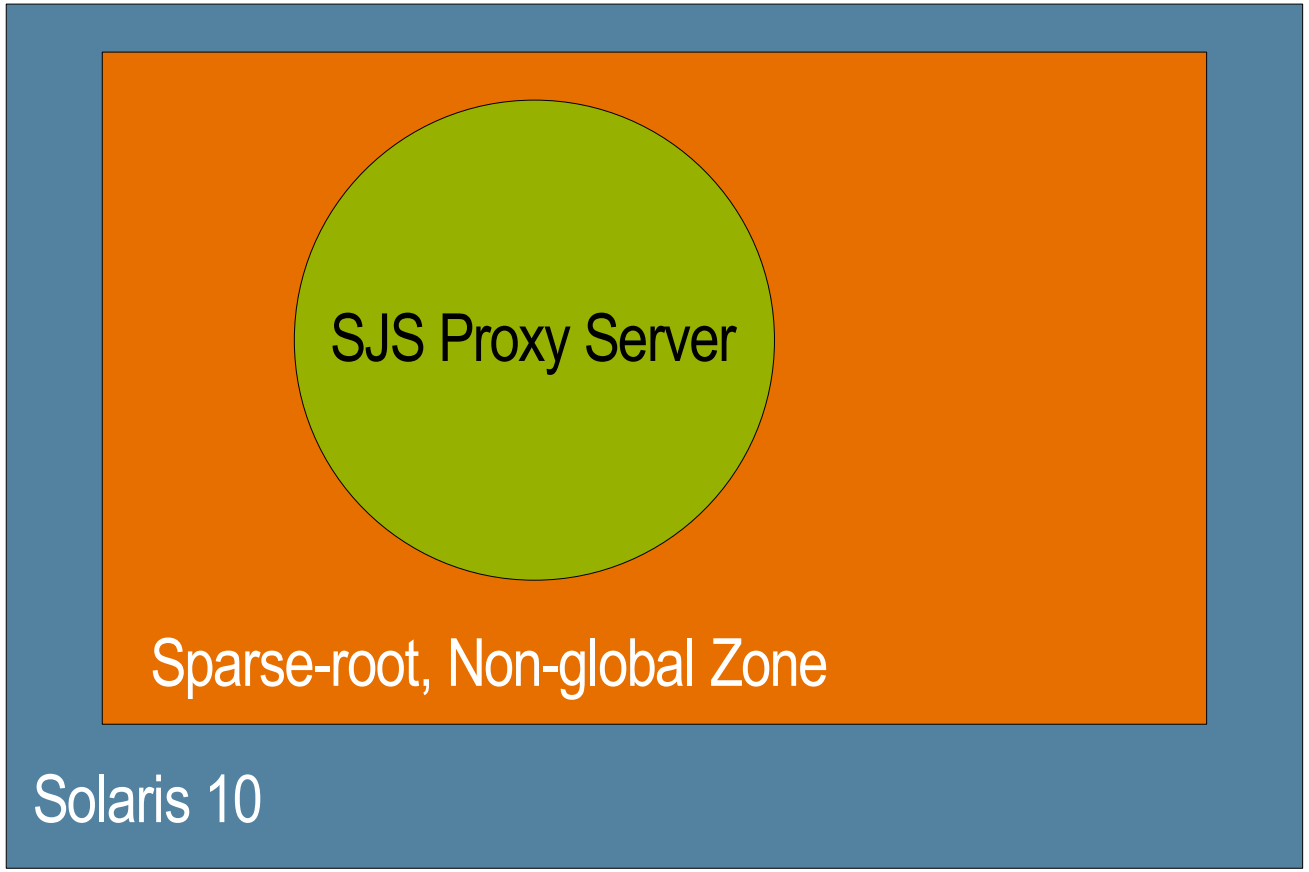
Secure Component

Composite Security Pattern Example

Secure Component

Secure Execution Container

Secure Component



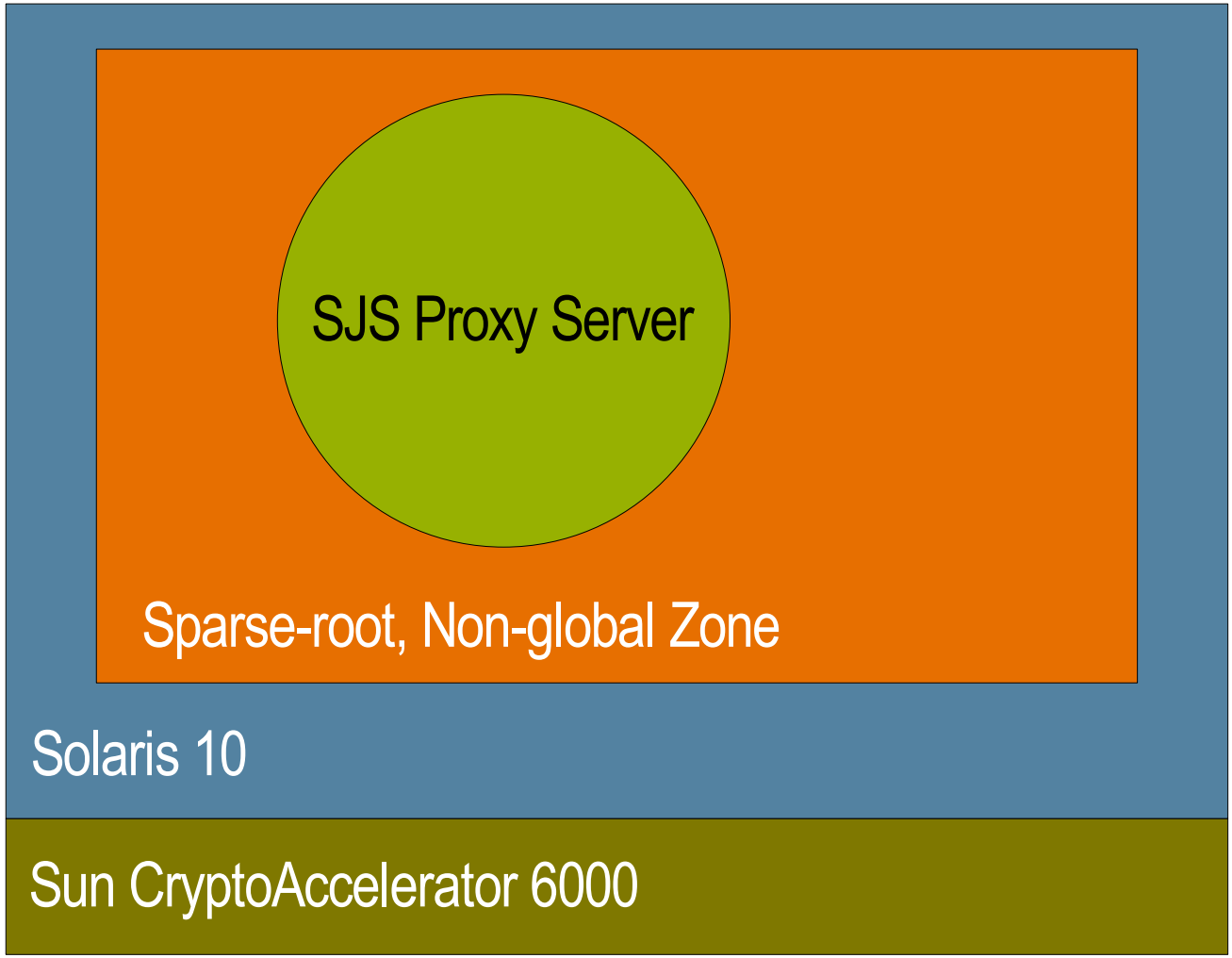
Composite Security Pattern Example

Secure Component

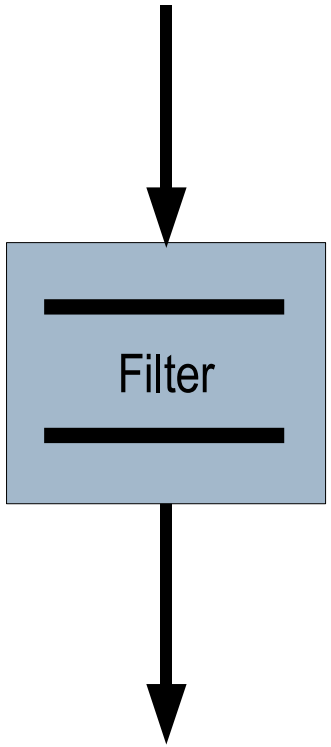
Secure Execution Container

Secure Component

Secure Execution Container

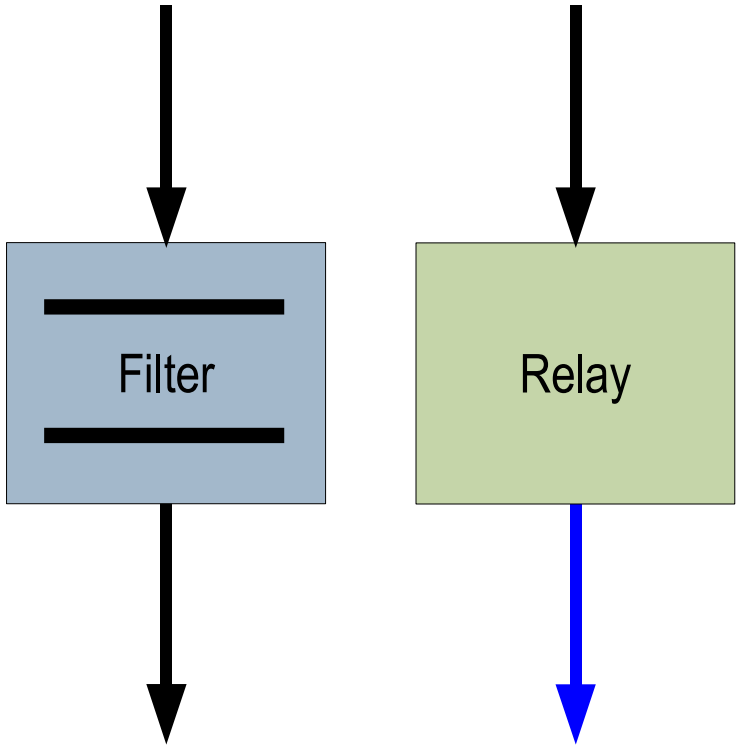


Additional Security Pattern Examples



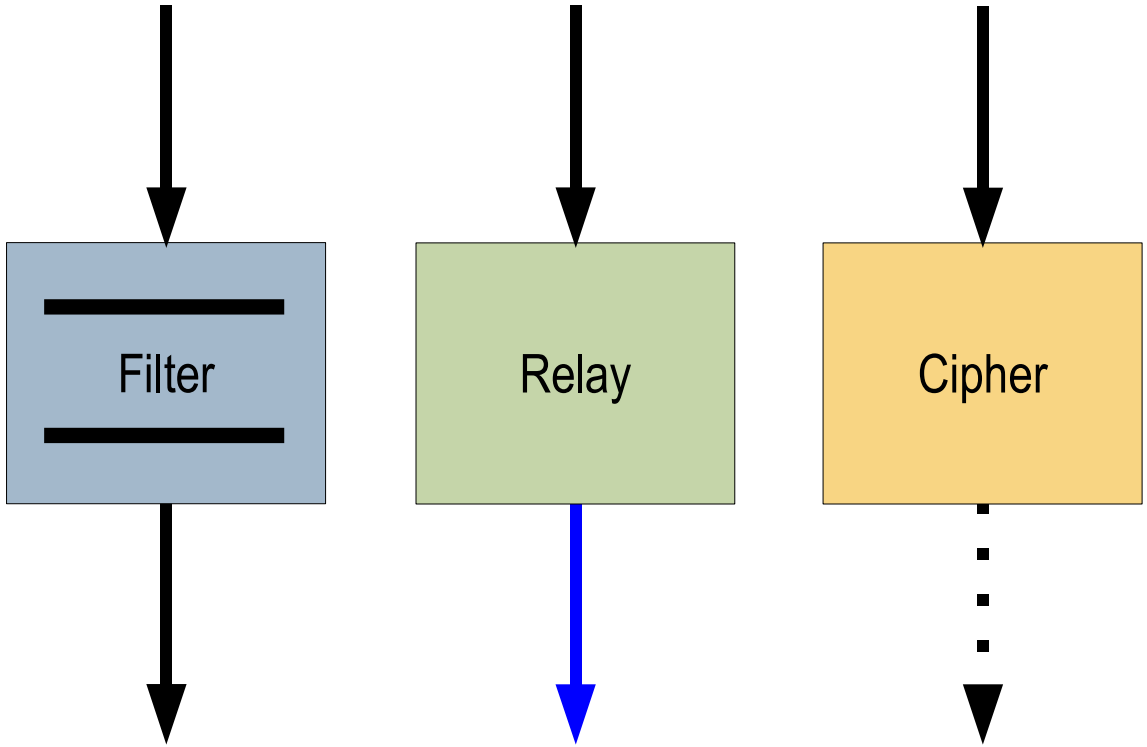
Each of these patterns inherit a set of core capabilities from the Secure Component and Secure Execution Container building blocks.

Additional Security Pattern Examples



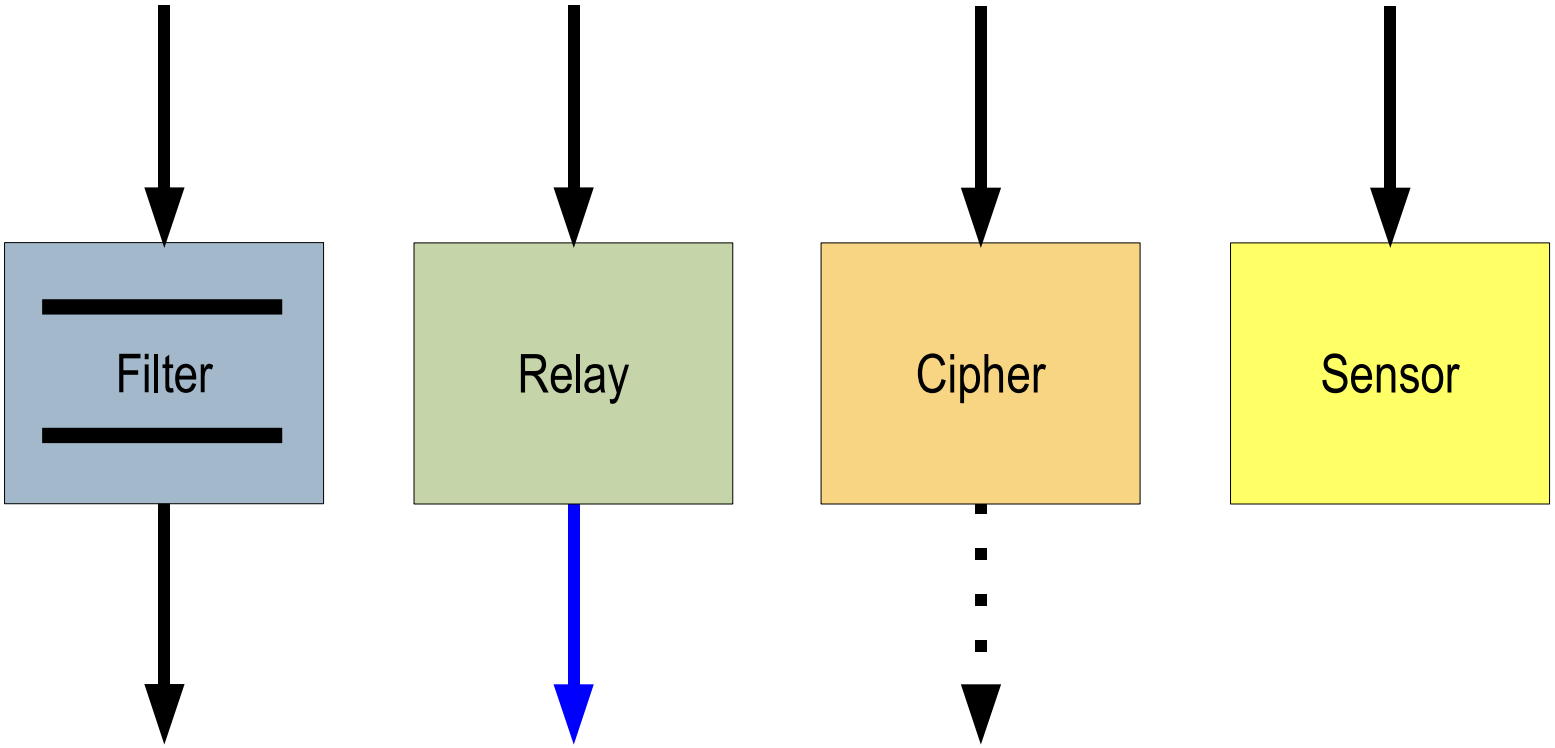
Each of these patterns inherit a set of core capabilities from the Secure Component and Secure Execution Container building blocks.

Additional Security Pattern Examples



Each of these patterns inherit a set of core capabilities from the Secure Component and Secure Execution Container building blocks.

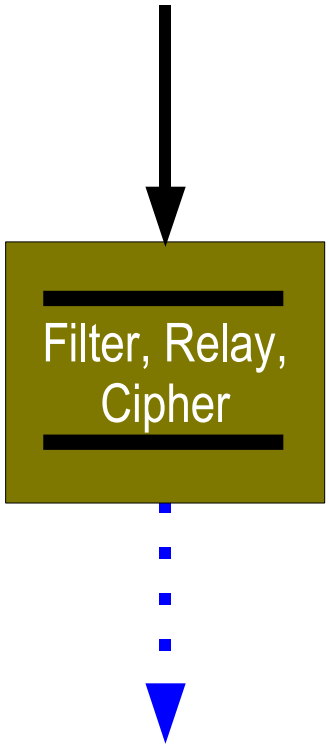
Additional Security Pattern Examples



Each of these patterns inherit a set of core capabilities from the Secure Component and Secure Execution Container building blocks.

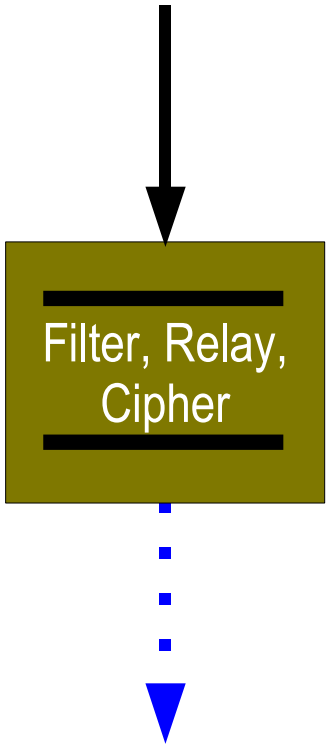
Composite Security Pattern Strategy

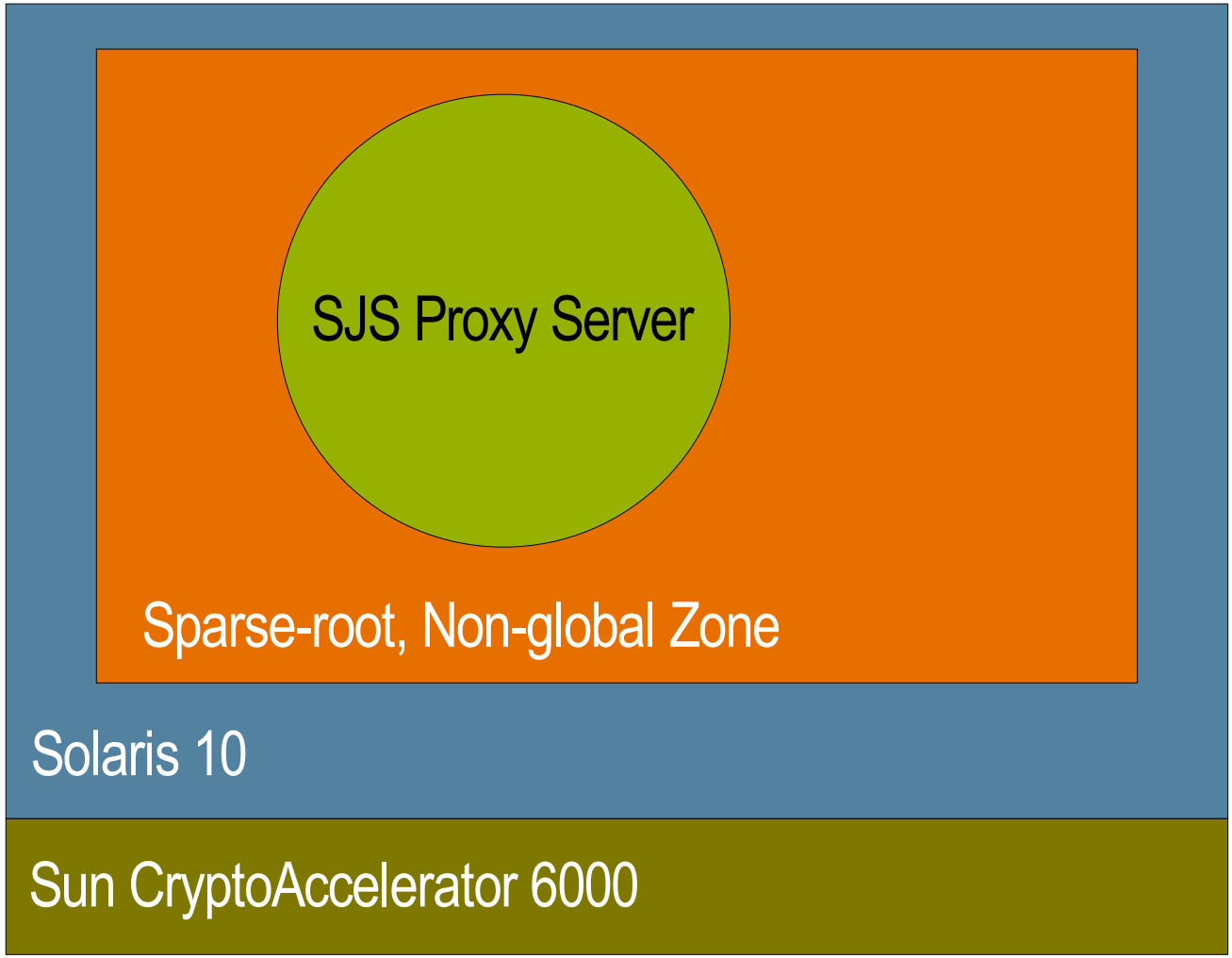
Secure (Reverse) Proxy



Composite Security Pattern Strategy

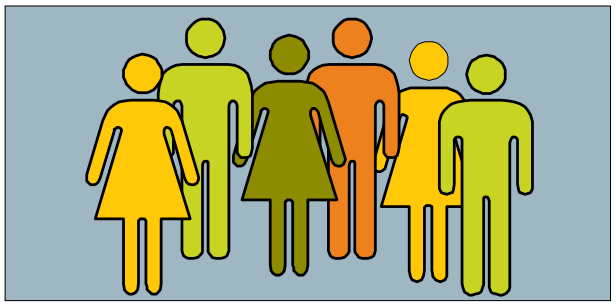
Secure (Reverse) Proxy



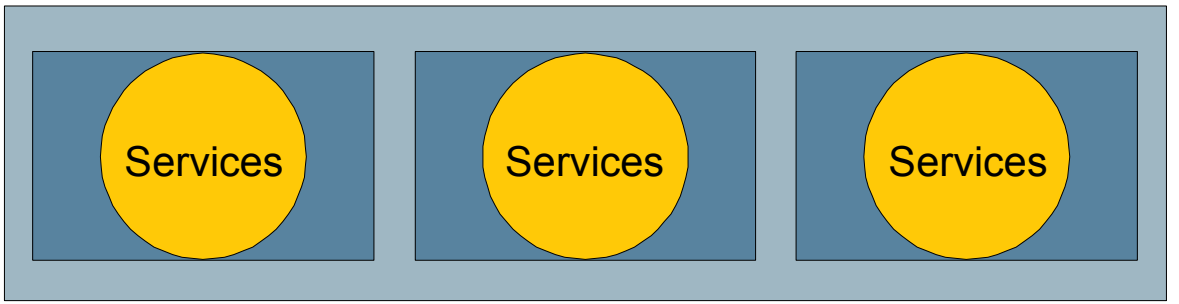


Secure Network Enclaves

- Group Elements into Secure Network Enclaves
  - > Based on community, services, threat profile, etc.
  - > Leveraging well-defined interfaces and access policies



Community Enclaves

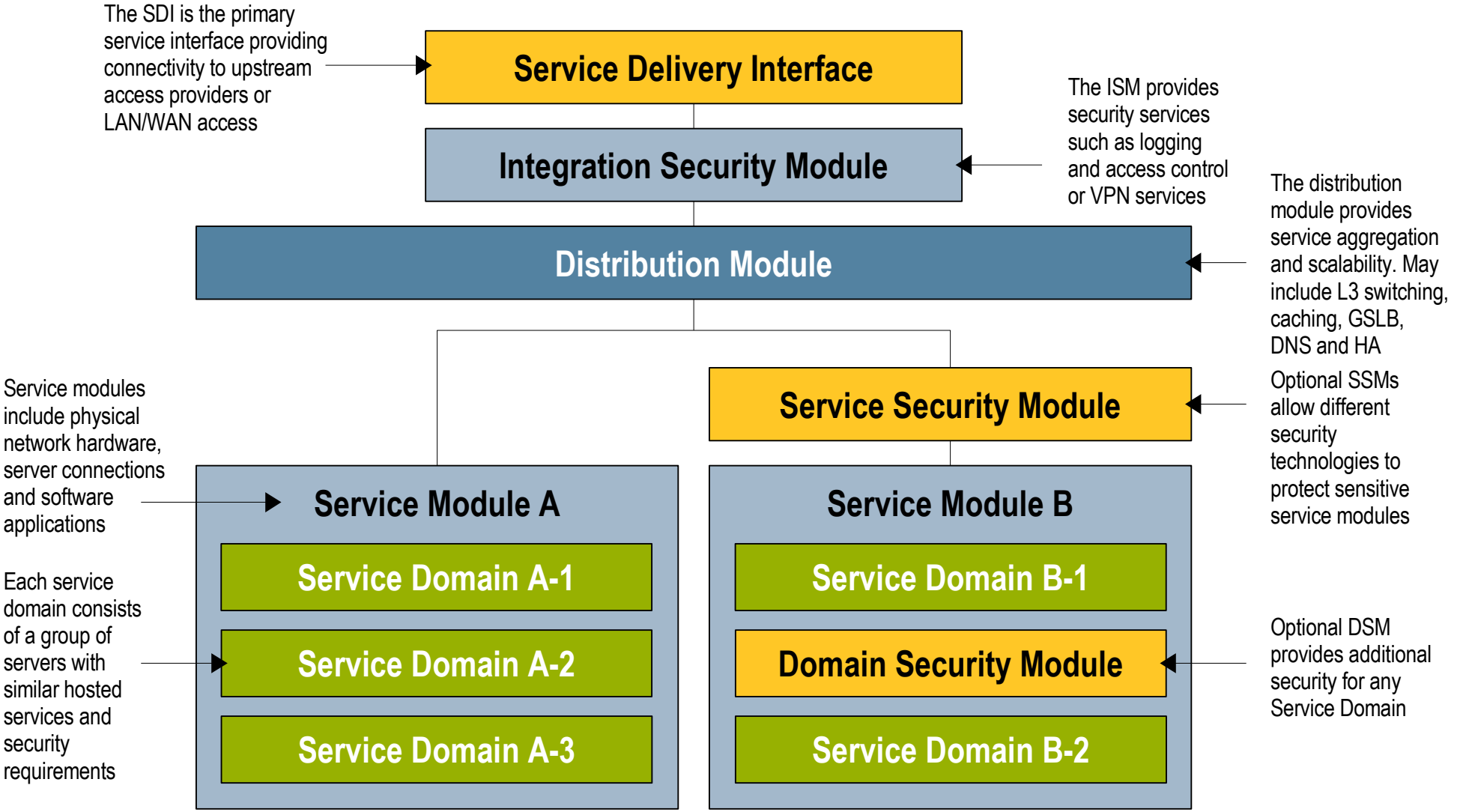


System or Service Enclaves

- Sun Service Delivery Network is the preferred instantiation
  - > Applied to both communities and services

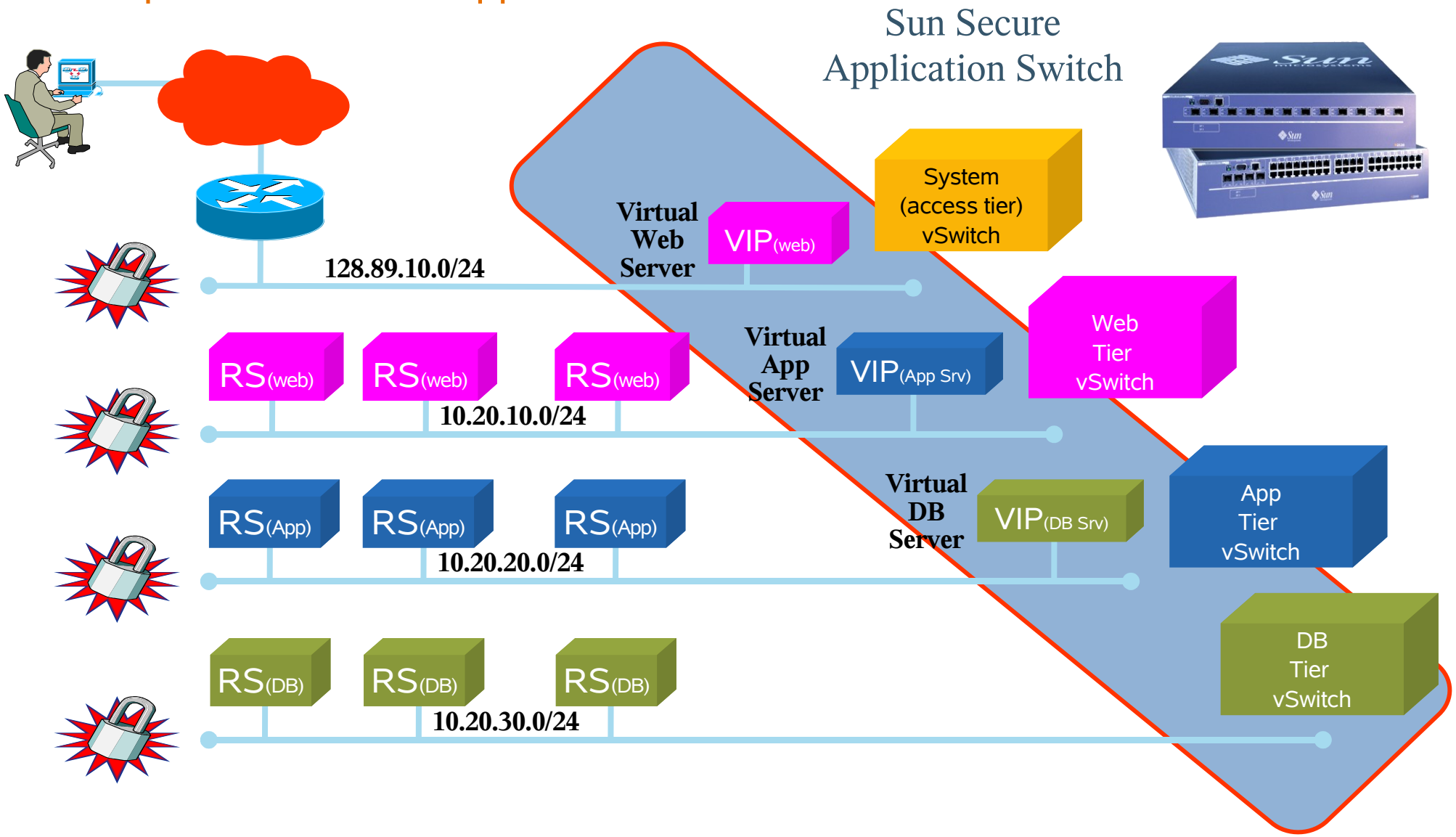
Secure Network Enclaves

Example: Sun Service Delivery Network (SDN) Architecture

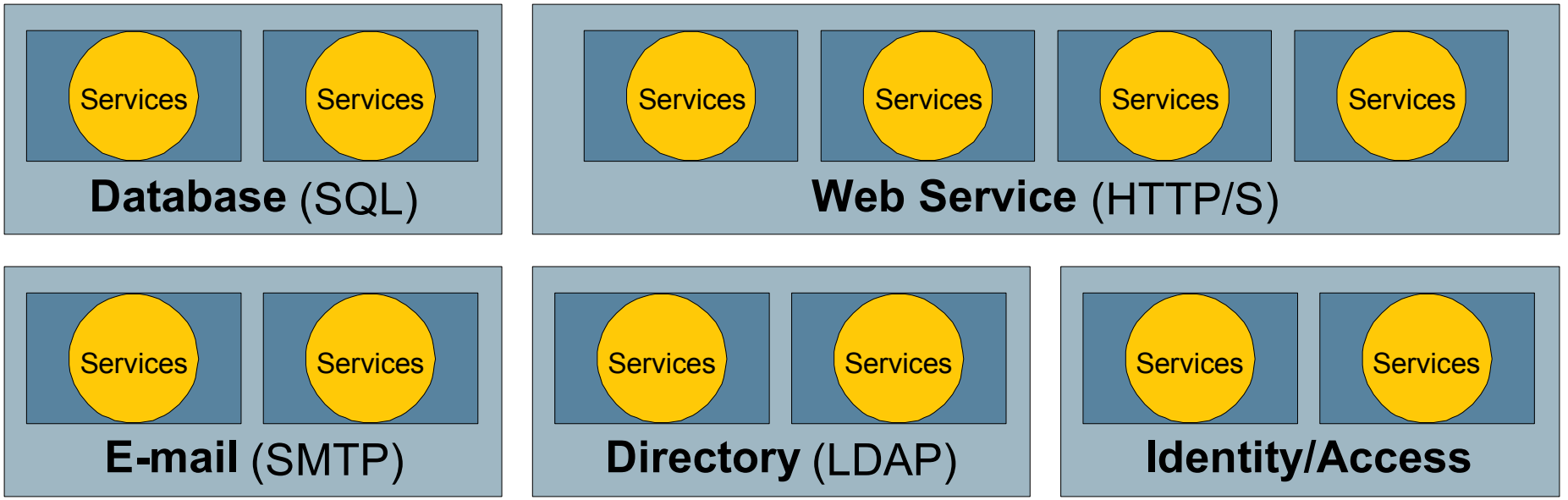


Secure Network Enclaves

Example: Sun Secure Application Switch



Shared Service Infrastructure

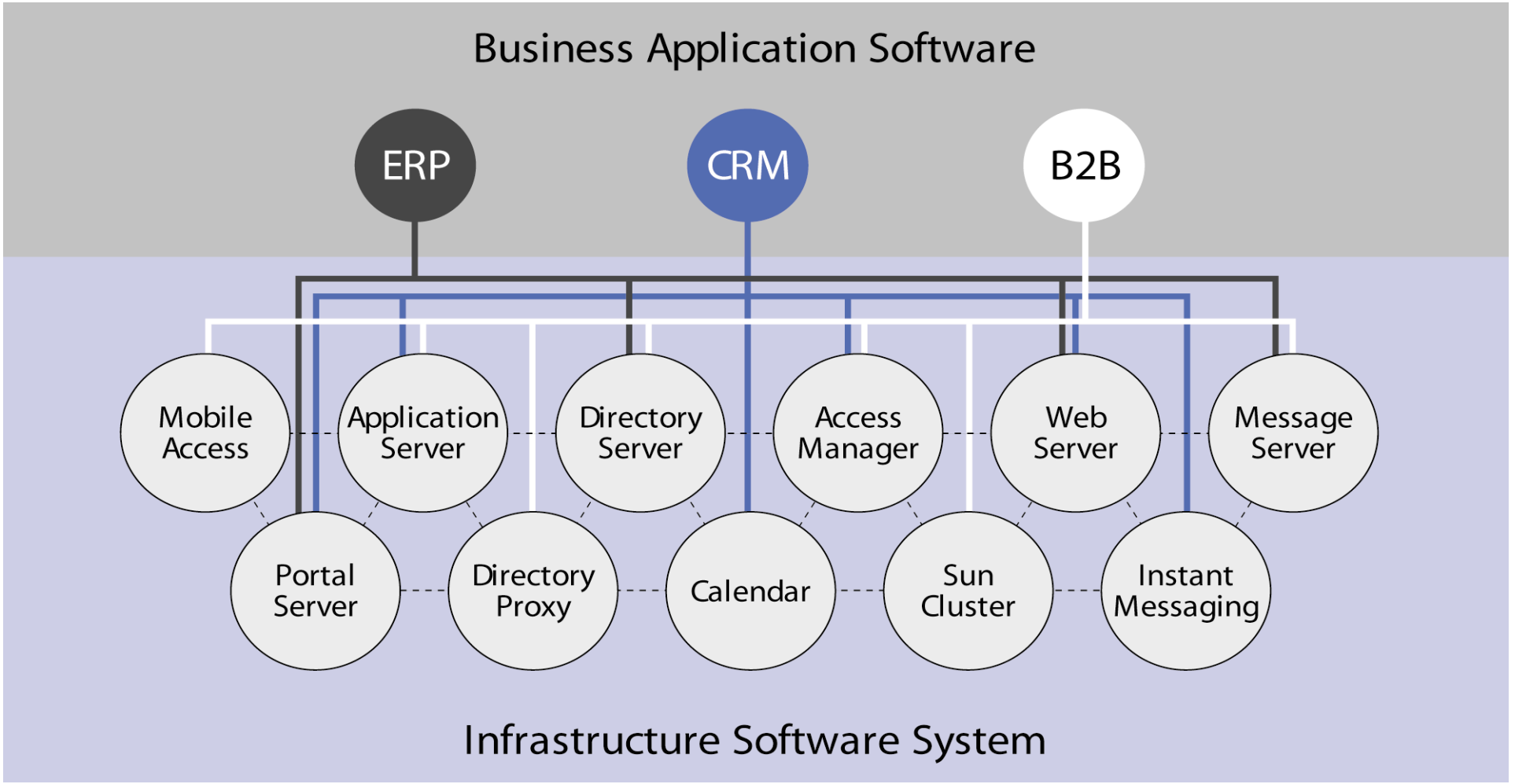


Why Shared Services?

- Supports consolidation and utility strategies.
  - > Fewer versions of software to purchase, install, support, backup, restore, maintain – reduced costs!
  - > Improves utilization, efficiency and cost of ownership.
- Leverages “standards” and automation.
  - > Improves agility, time to deployment and availability.
  - > Simplifies troubleshooting and root cause analysis.
- Improves the overall IT security lifecycle.
  - > Easier to know what you have, where it is, and who has access.
  - > Fewer versions of software to secure, maintain and monitor.
  - > Improves ability to identify and mitigate discovered vulnerabilities.

Shared Service Infrastructure

Example: Sun Java Enterprise System



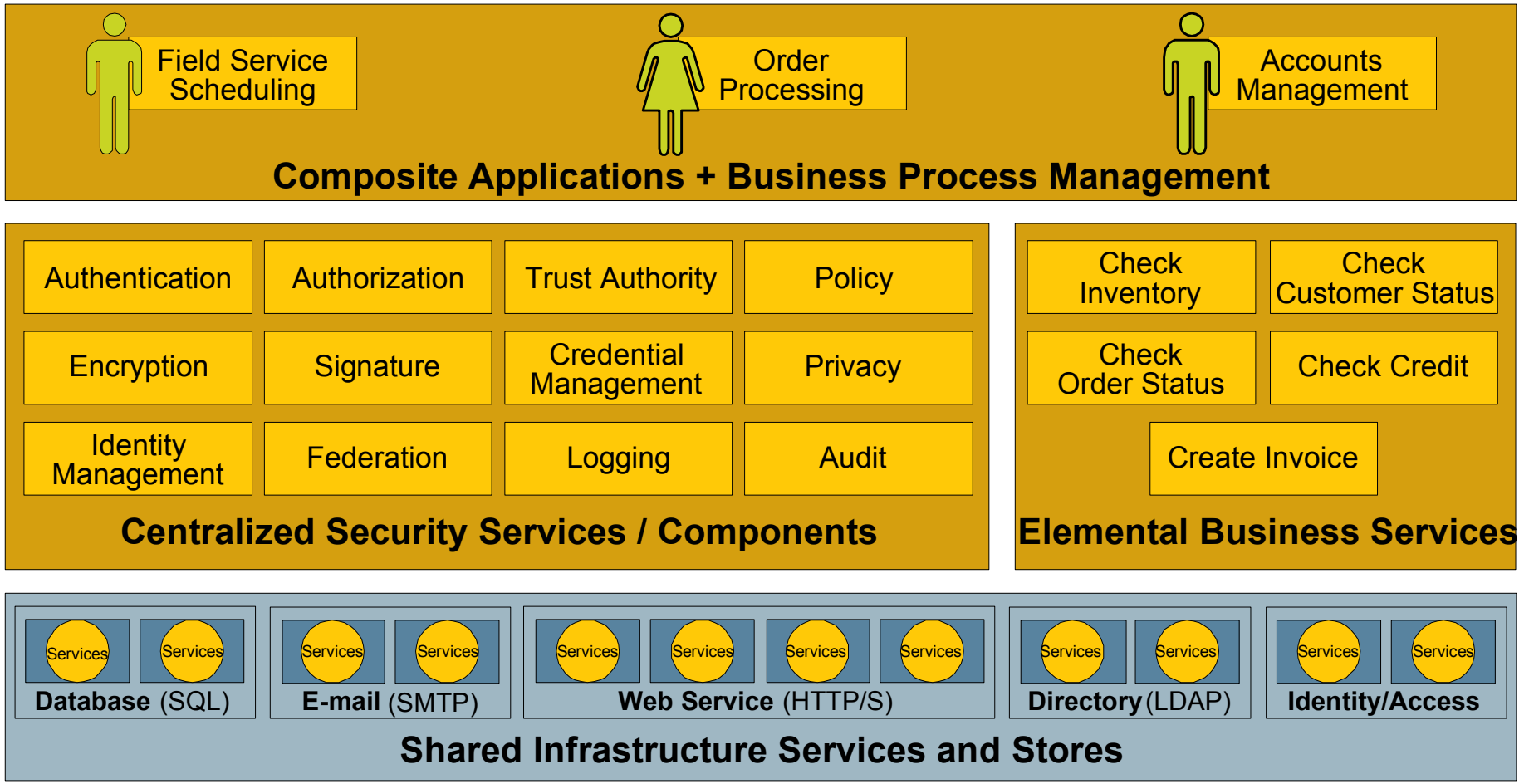
Automating Shared Service Infrastructure

Example: Sun N1 Service Provisioning System

Key Features

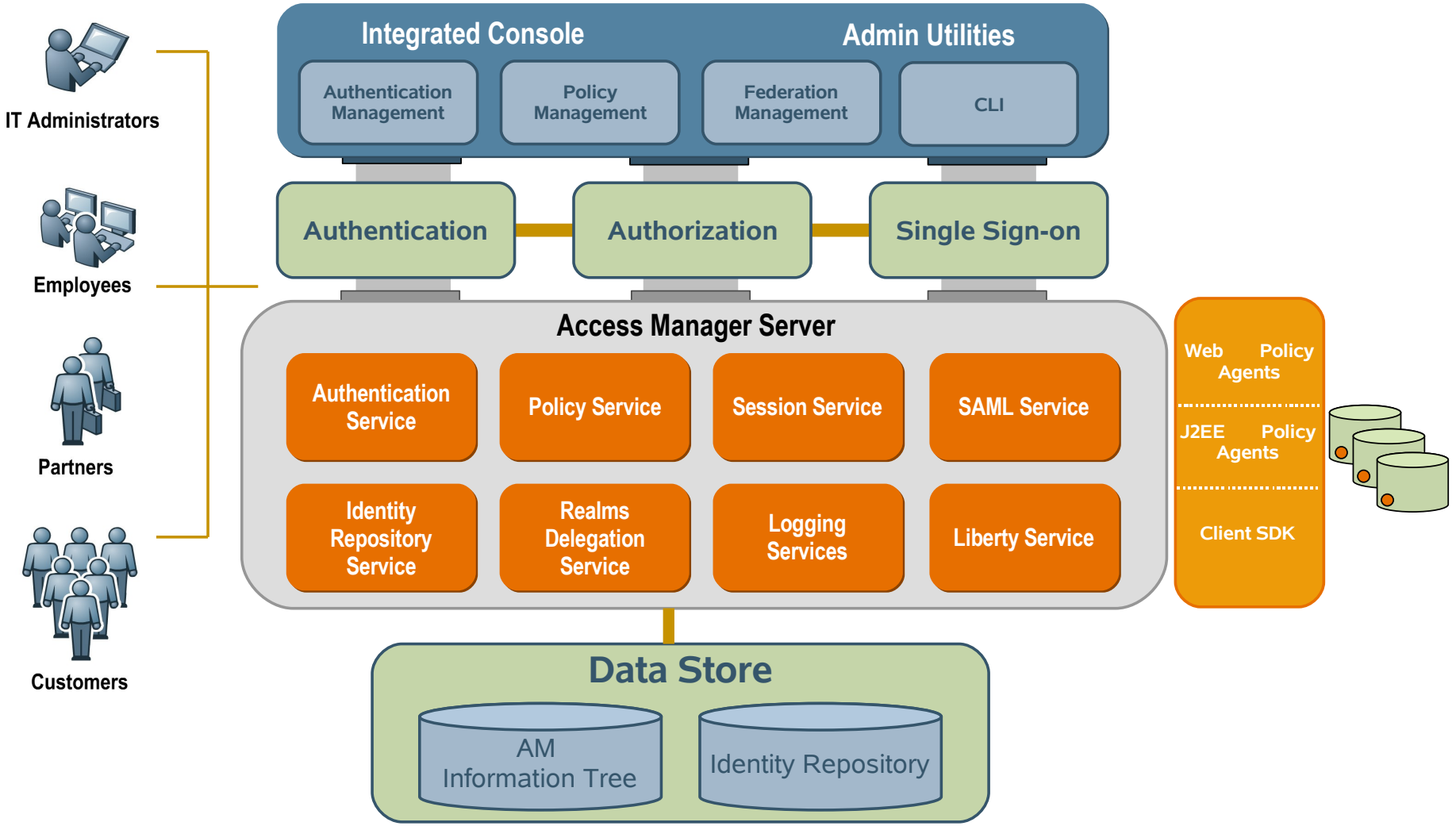
- Secure OS and Multi-Tier Application Provisioning
- Application Portfolio
- Configuration Comparison
- Version Control and Rollback
- Role-based Access Control

Shared Application Services



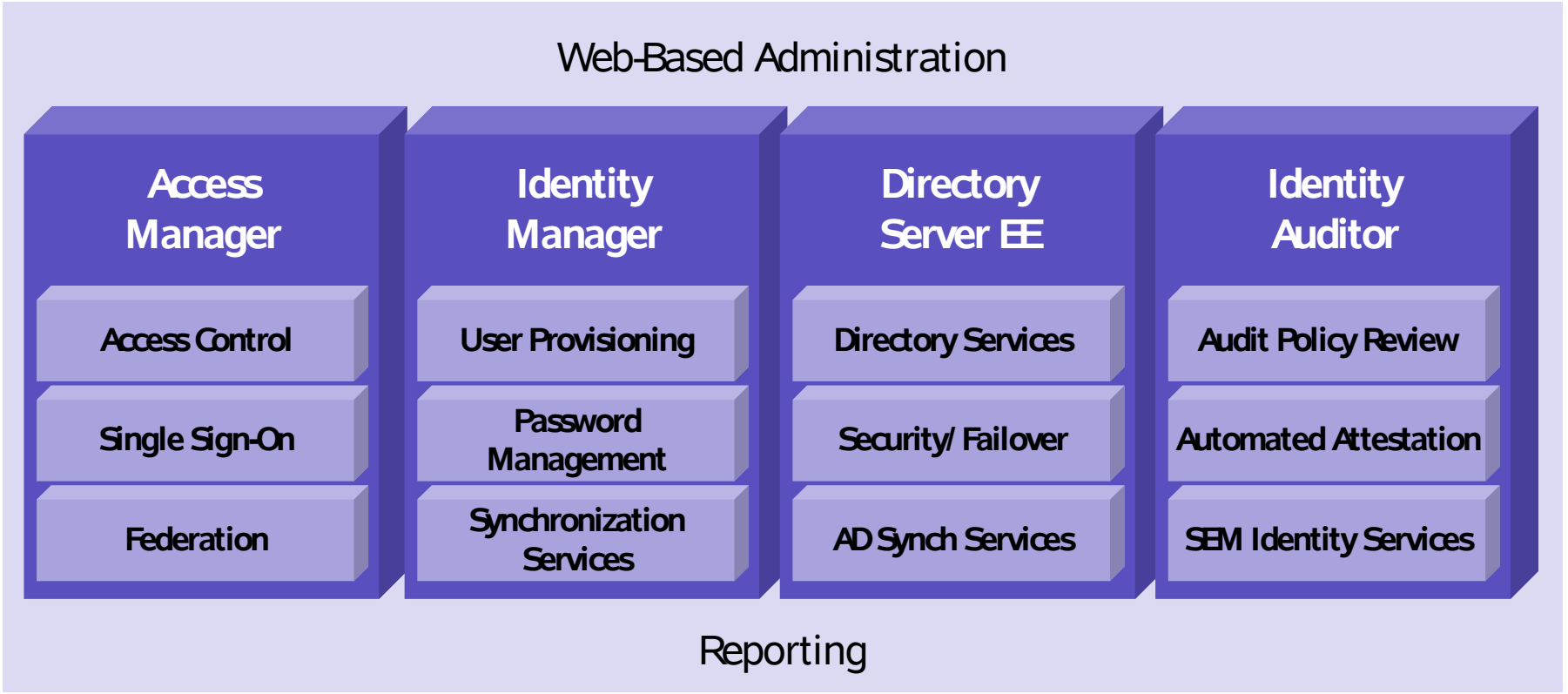
Shared Application Services

Example: Sun Java System Access Manager

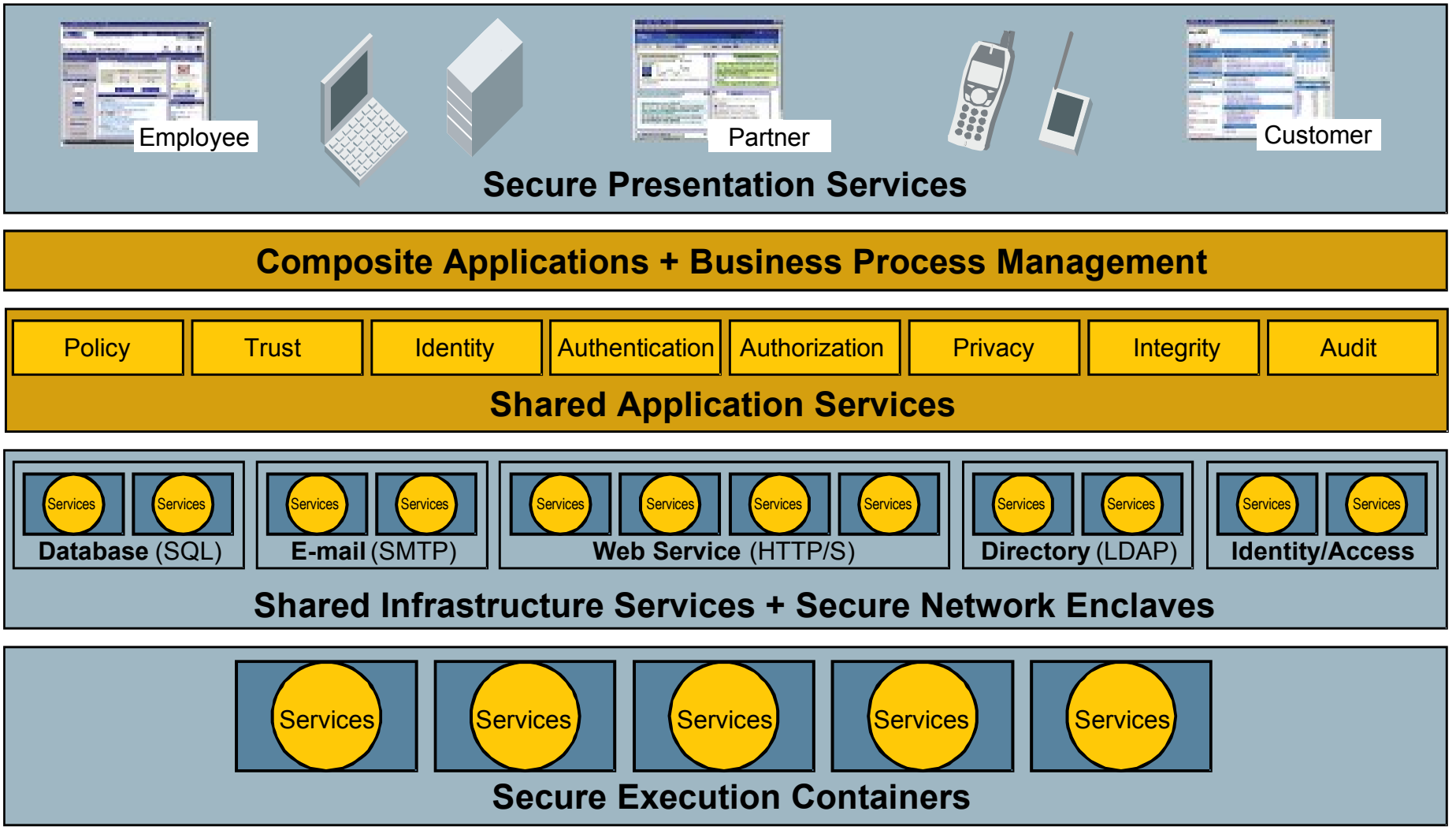


Unified Identity and Access Management

Bridging Both Worlds: Infrastructure and Applications



Secure Presentation Services



Secure Presentation Services

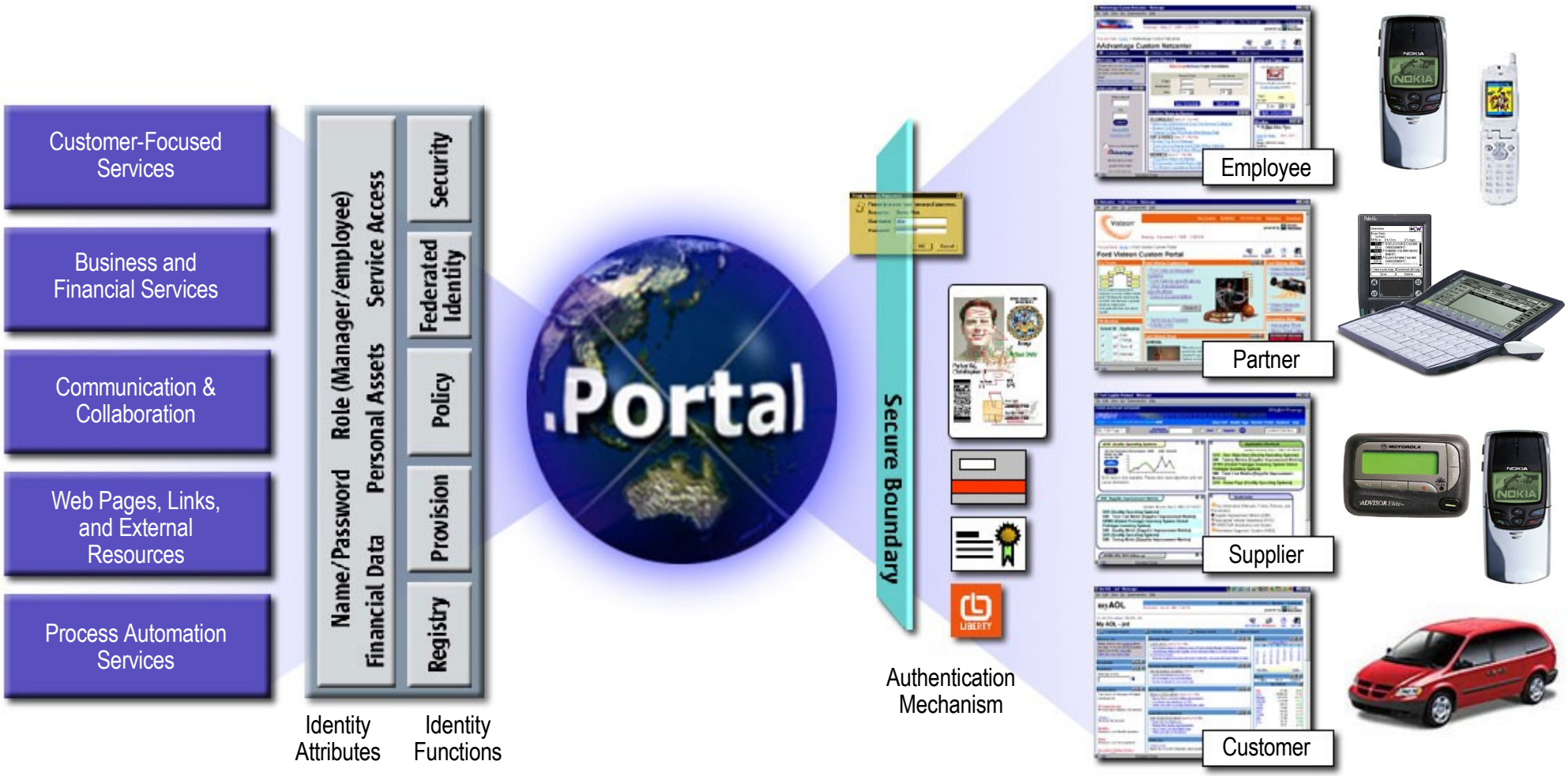
Example: Sun Java Enterprise System Portal Server

Data No Matter Where It Resides

Aggregated and Personalized

Securely Delivered to Targeted Communities

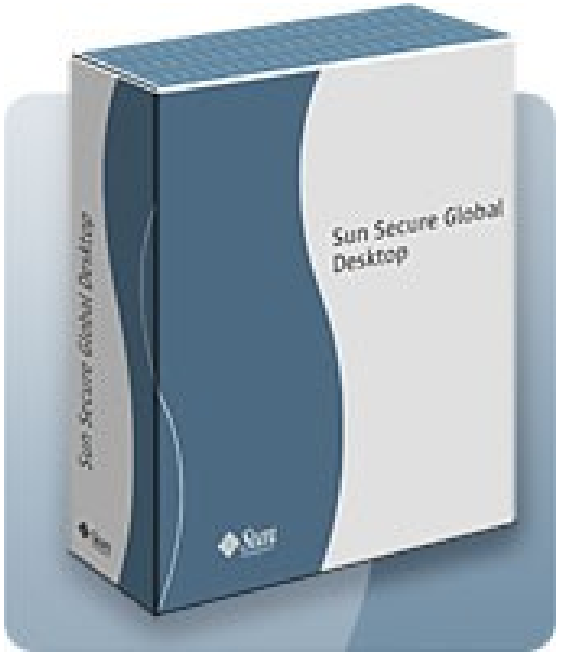
Via Any Device



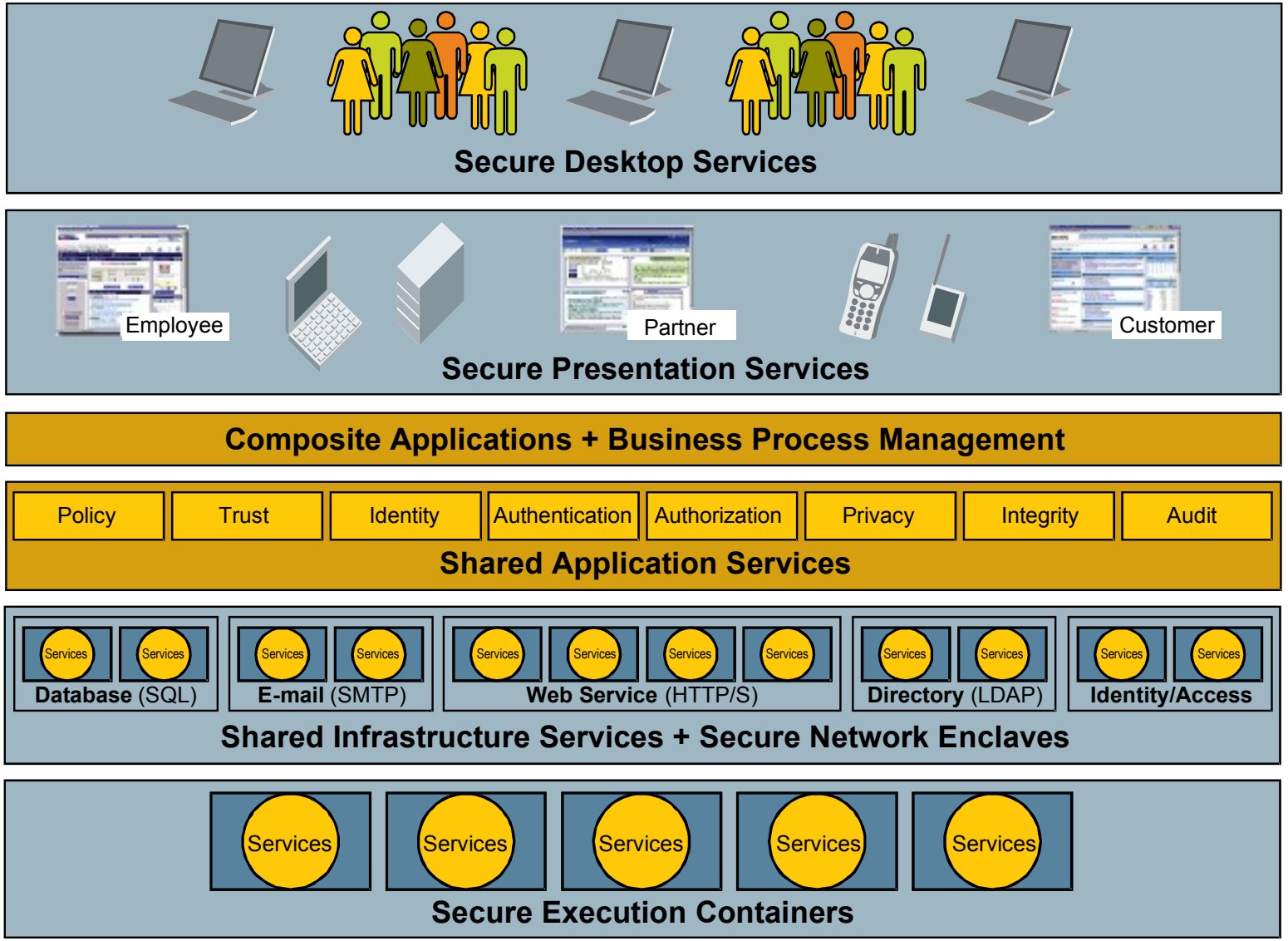
Secure Presentation Services

Example: Sun Secure Global Desktop

- Secure out of the box.
  - Strong, pluggable authentication.
  - Strong encryption.
  - n-Tier Architectural model.
  - Session shadowing and auditing.
- Centralized control of access to applications and services.
  - Access content from any application and deliver it to virtually any device or client.
  - Seamless access to Windows, Unix and Mainframe-based applications and services.
  - Allows centralized control of users, applications, printers and other resources.



Secure Desktop Services

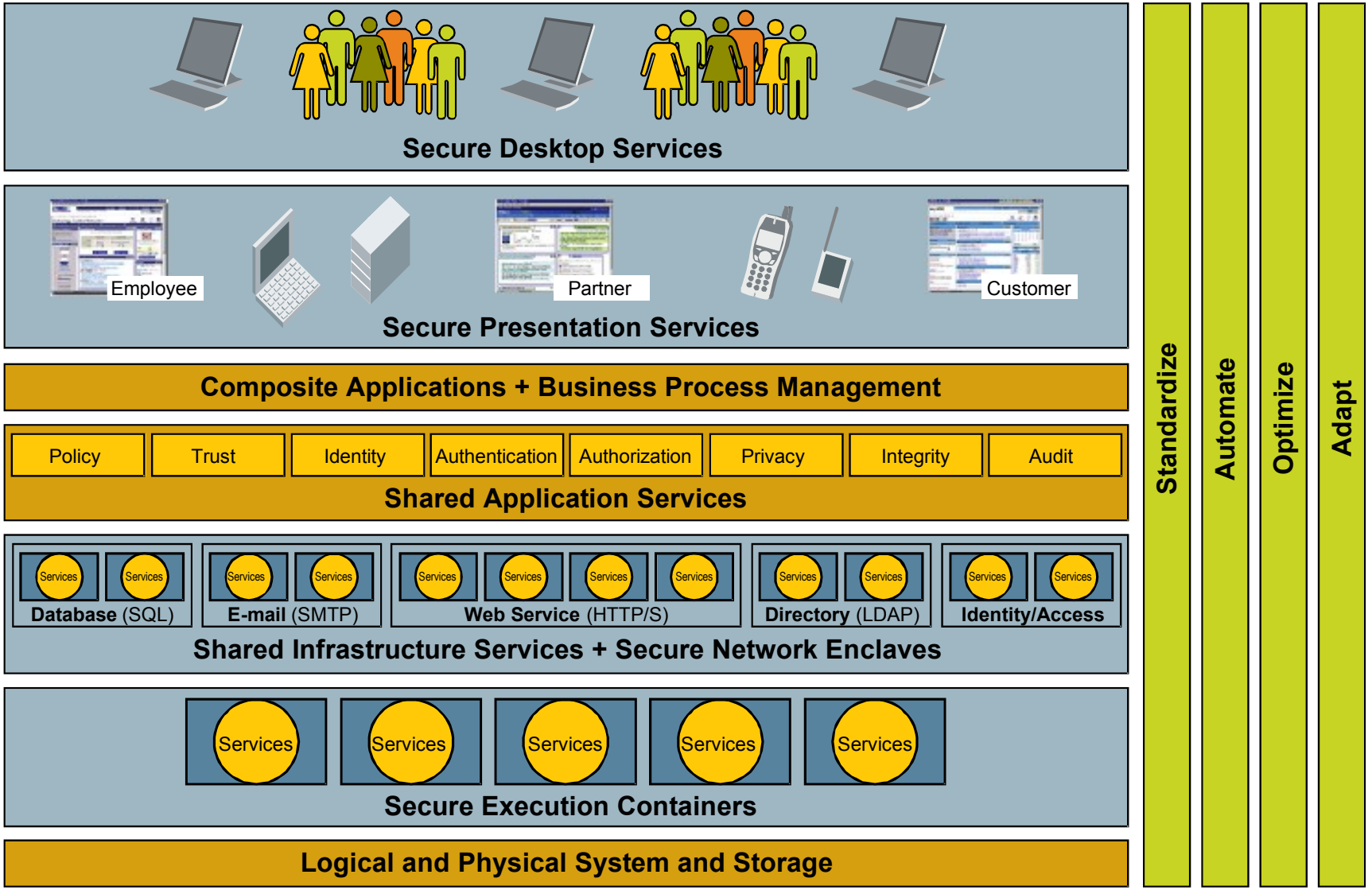


Secure Desktop Services

Example: Sun Ray

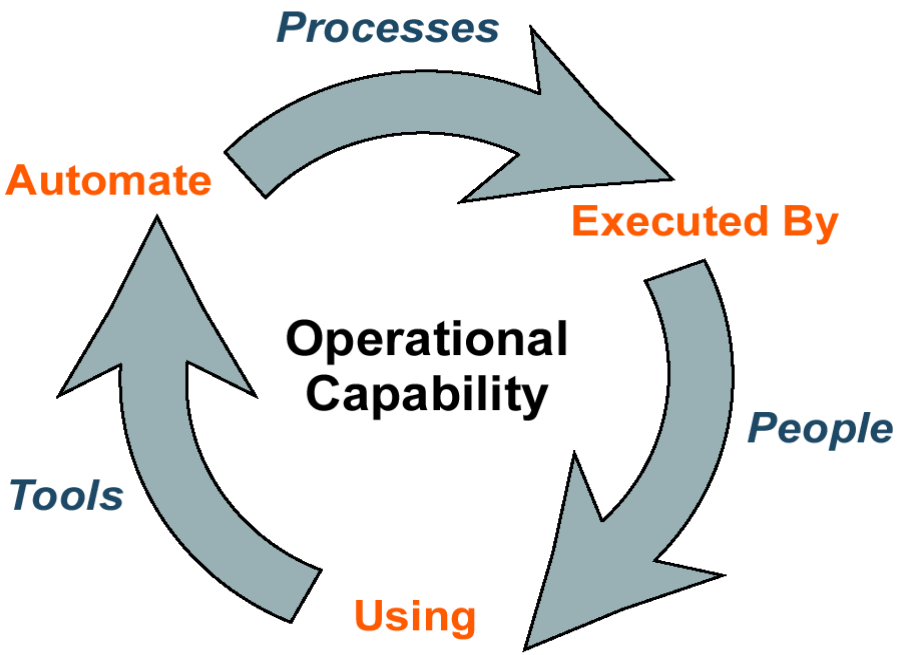
- Secure out of the box.
  - Zero client-side configuration.
  - No local software or state.
  - No viruses, worms, etc.
  - No moving parts to cause wear or break.
- Centralized users, data and services.
  - One administrator per every 2,000 desktops.
  - Complete user deployments in a day.
  - Operating system upgrades in months (not years!)
  - Application upgrades in days (not months!)
- Significant cost savings.
  - No annual hardware refresh costs.
  - Low power consumption (<20W)

Continuous Improvement



Continuous Improvement

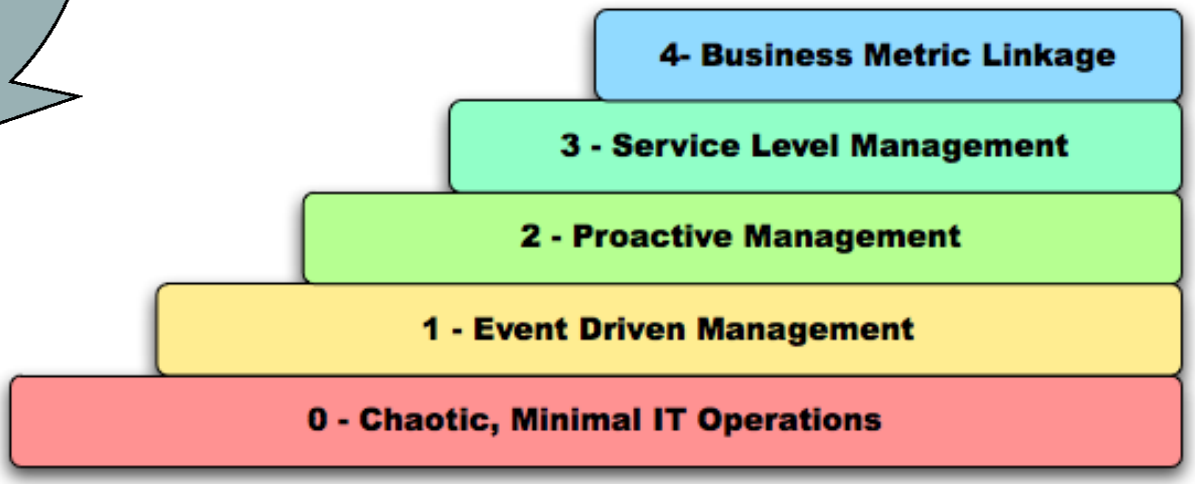
Example: Sun Operations Management Capability Model



ITIL
The IT Infrastructure Library

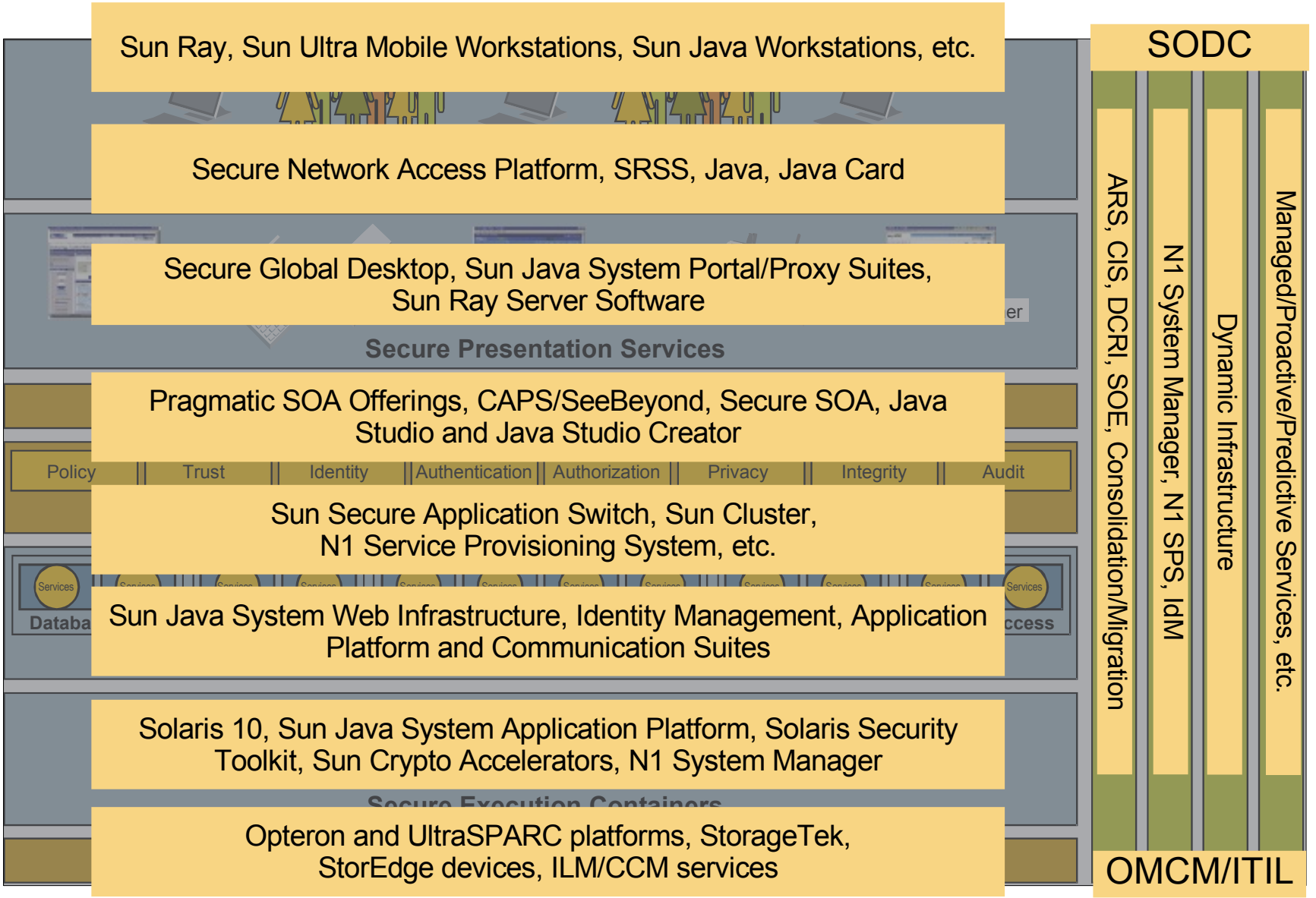
Industry Standard Best Practices

SIX SIGMA

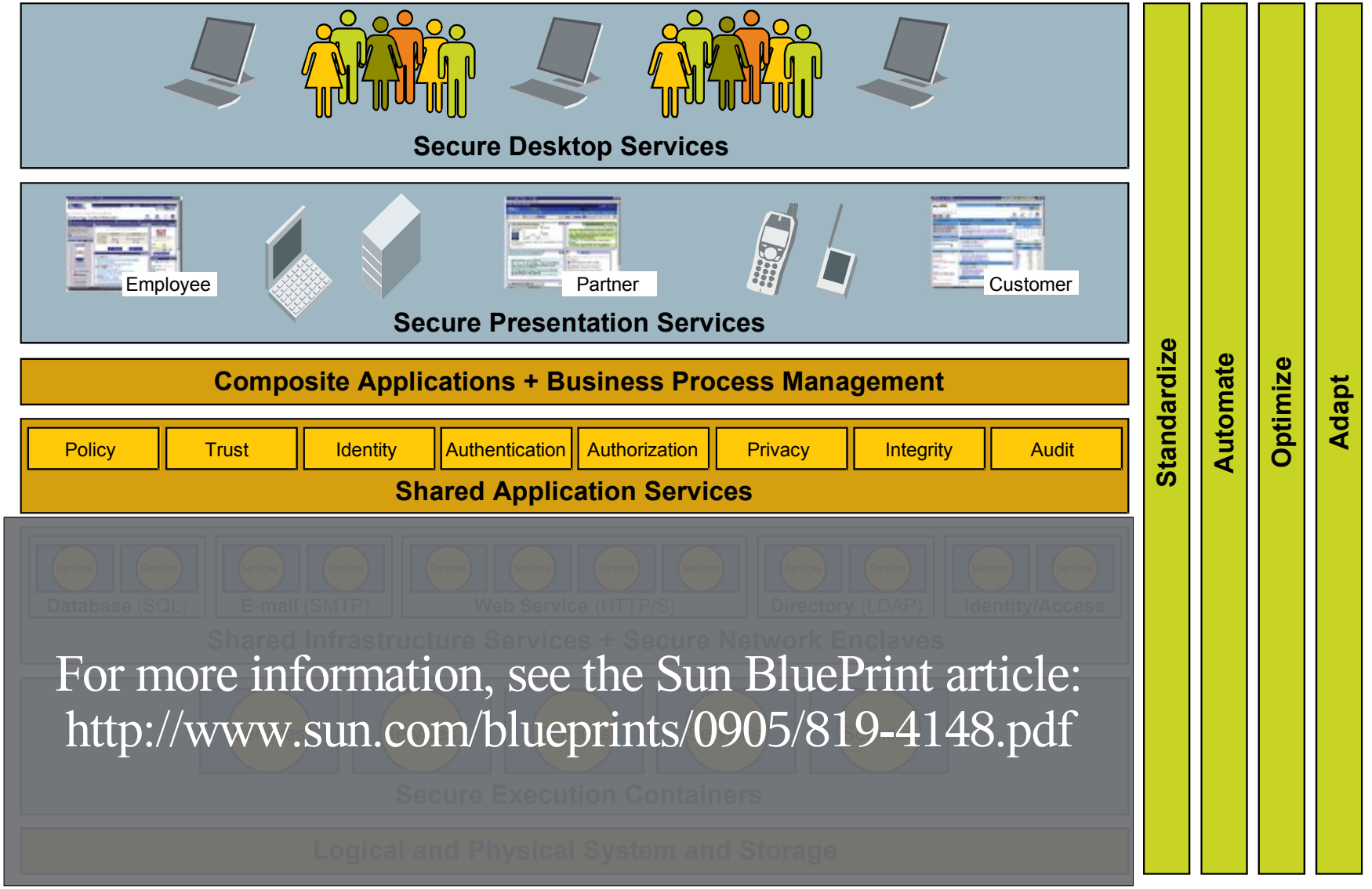


Operations Maturity Model

Sample Sun Product/Service Mapping

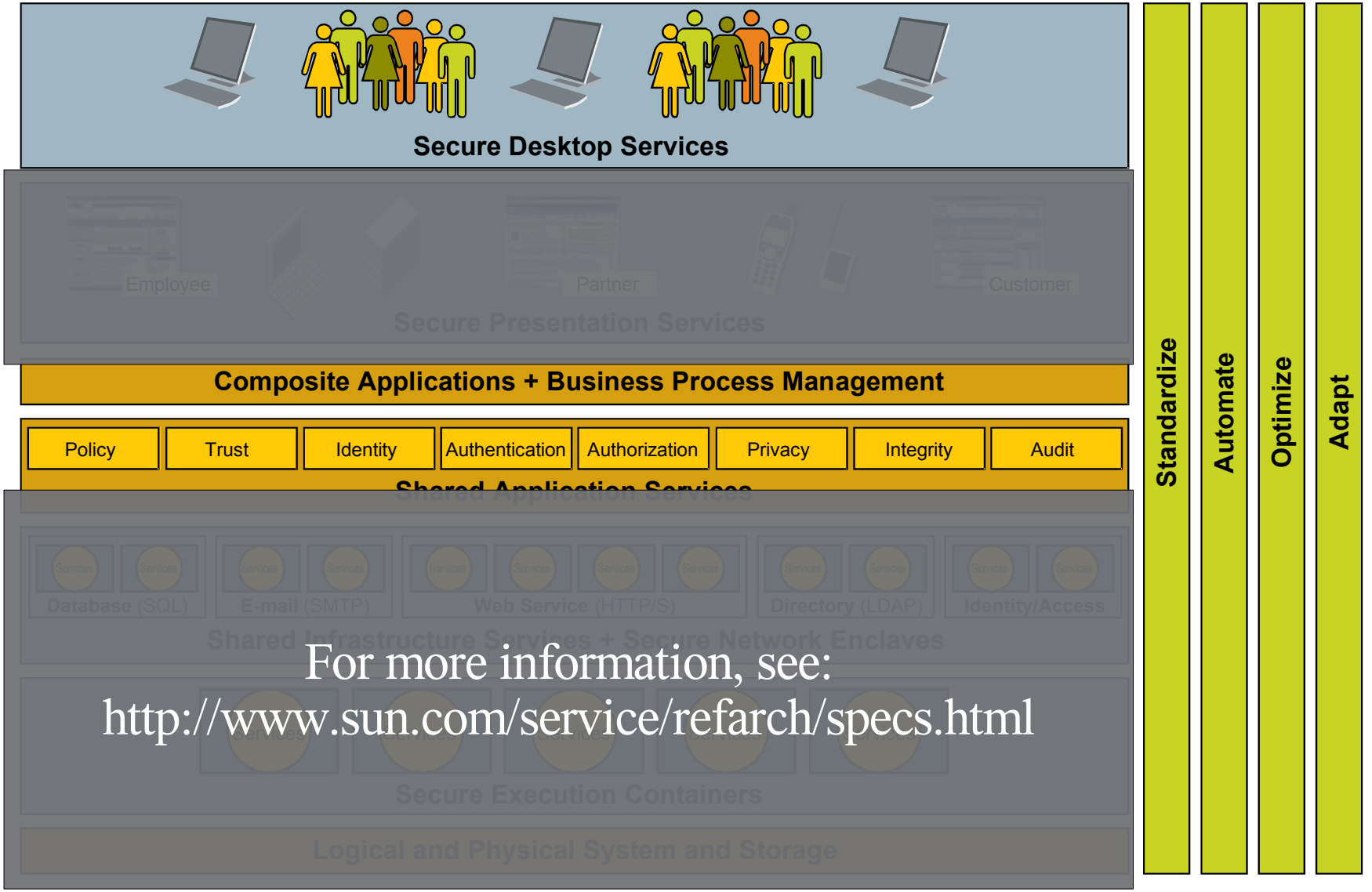


Service Delivery Network (SDN)



For more information, see the Sun BluePrint article:
http://www.sun.com/blueprints/0905/819-4148.pdf

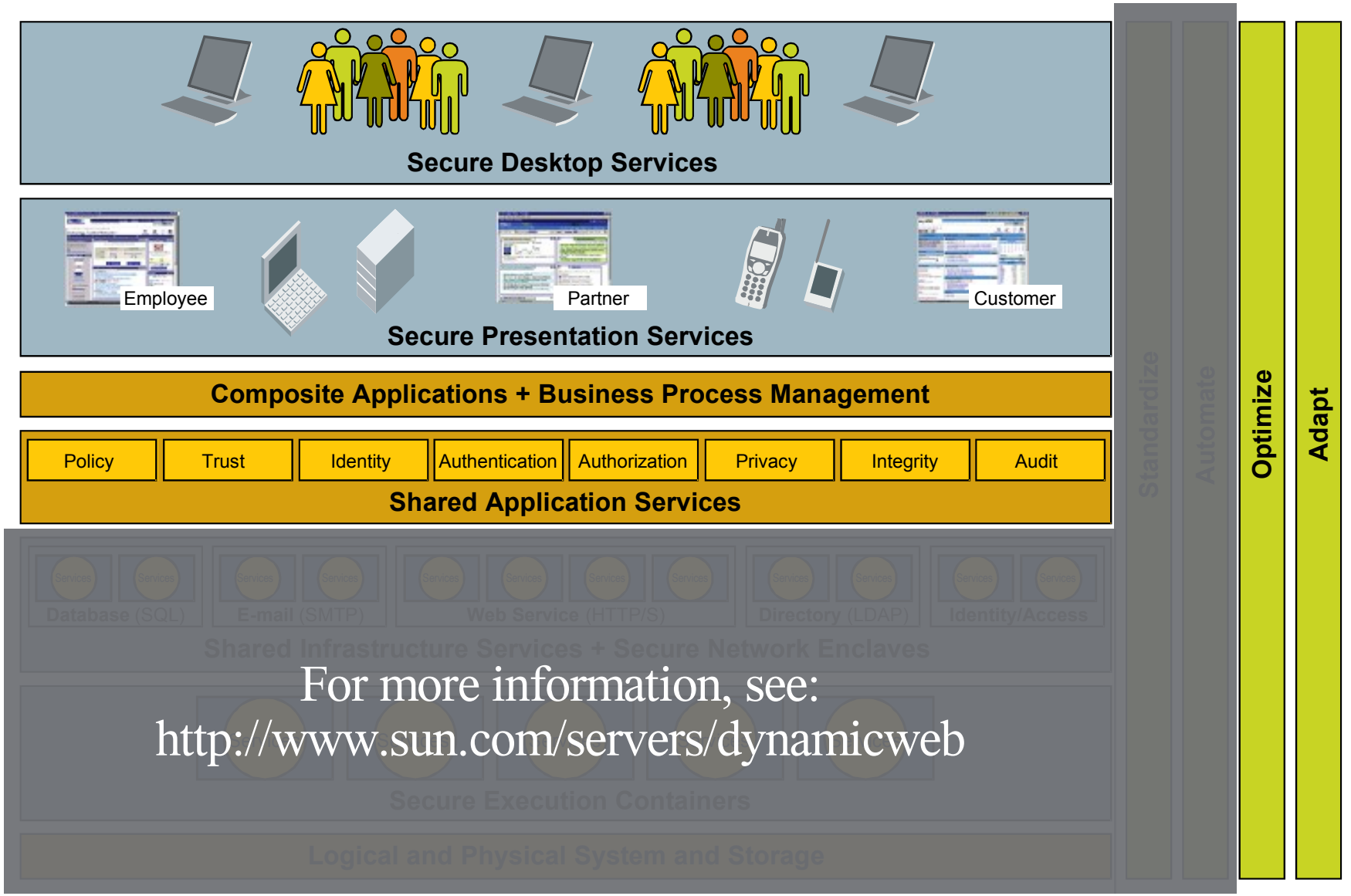
Portal Services Reference Configuration



For more information, see:
http://www.sun.com/service/refarch/specs.html

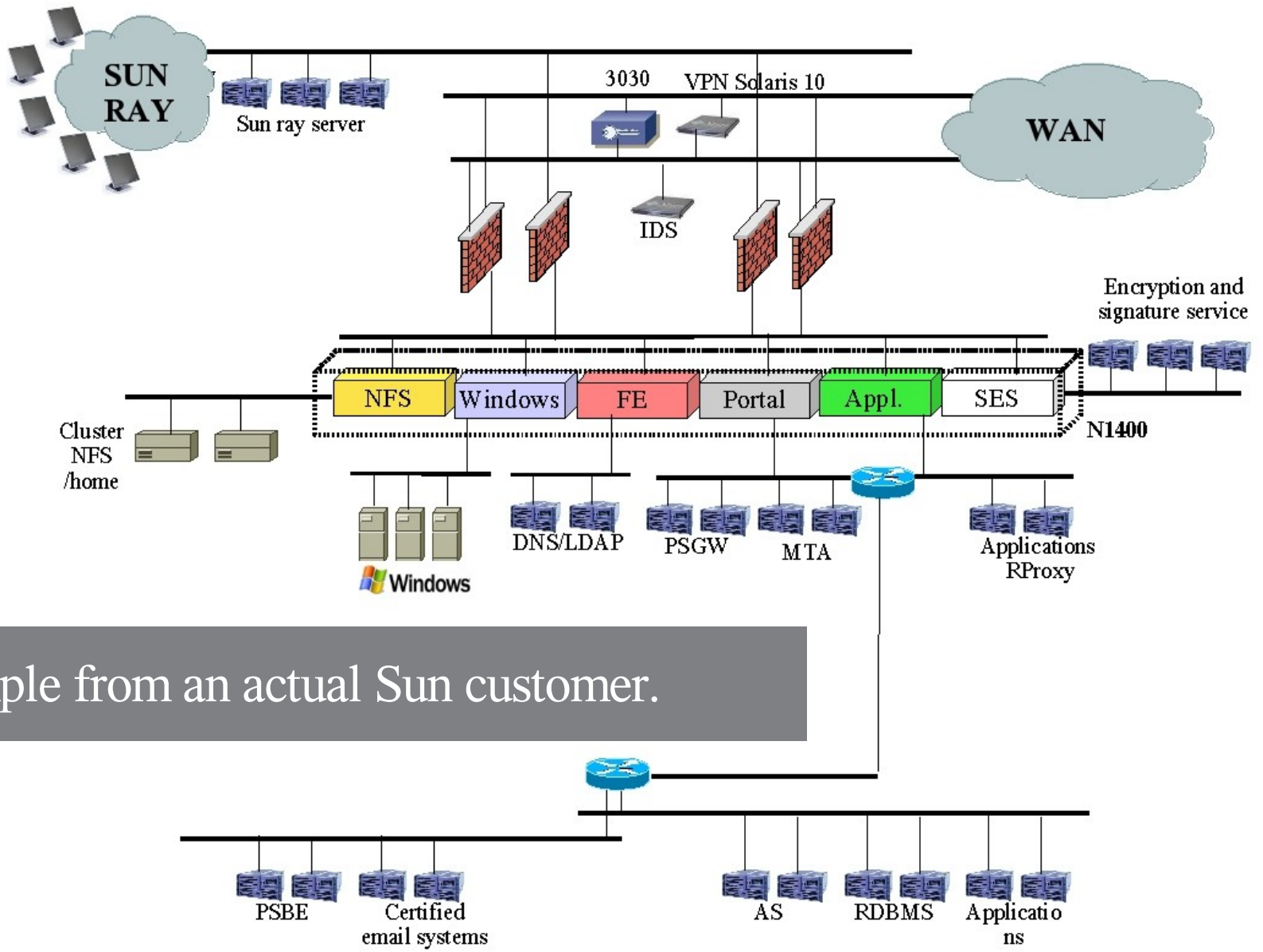
Secure Network Access Platform (gSNAP-2)

Dynamic Infrastructure for Web Services (DI)



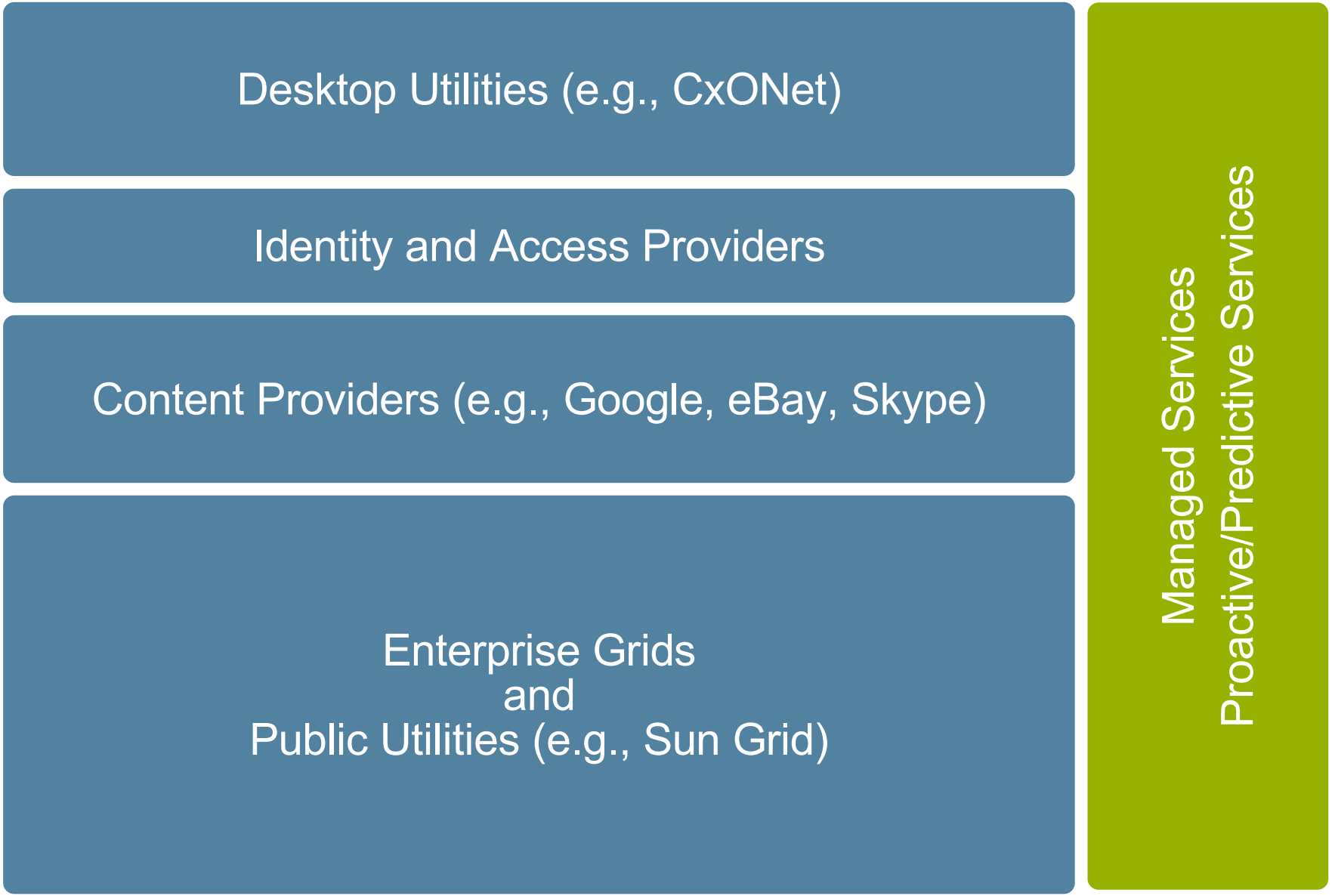
For more information, see:
http://www.sun.com/servers/dynamicweb

A Sun Systemic Security Success!

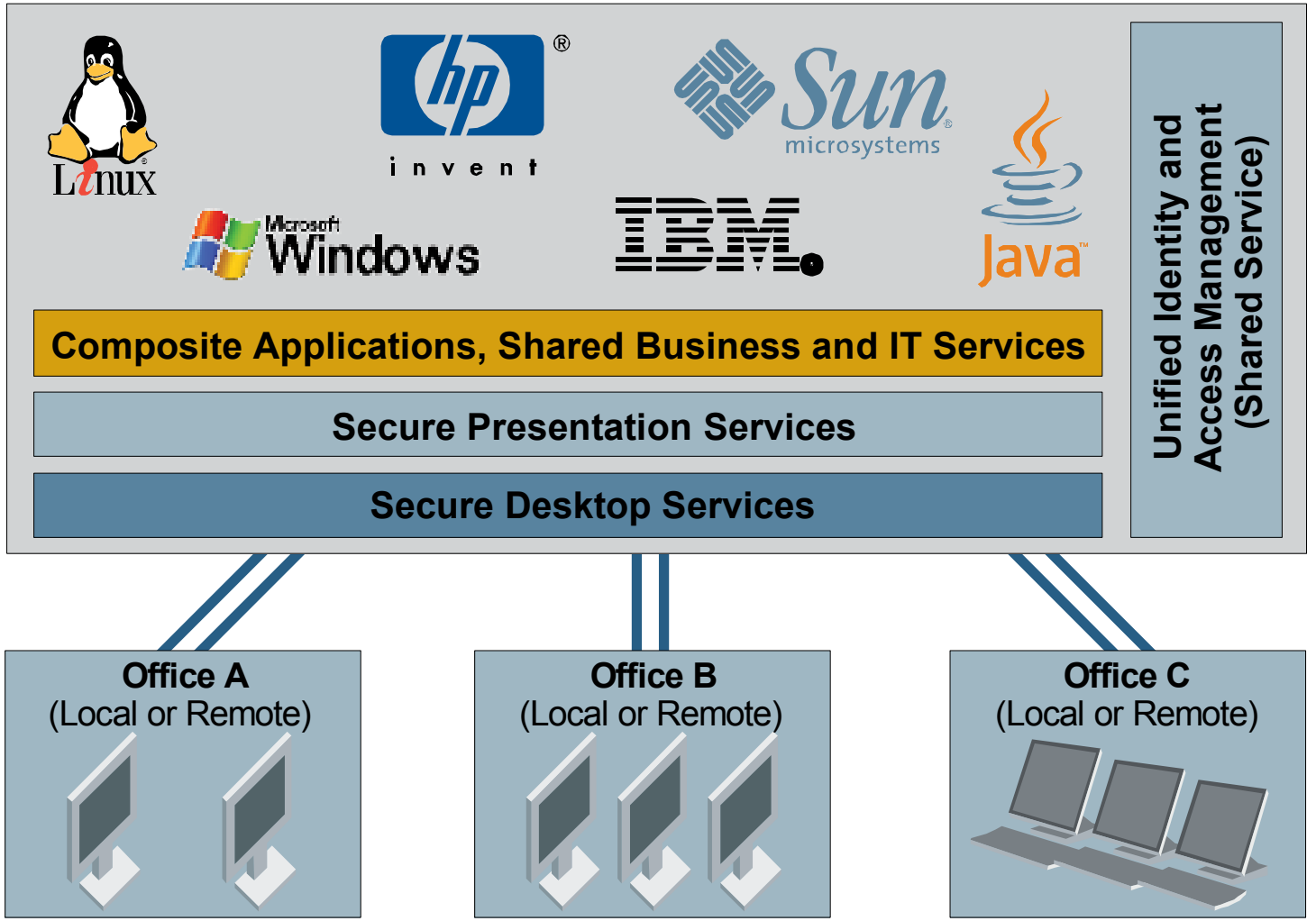


Example from an actual Sun customer.

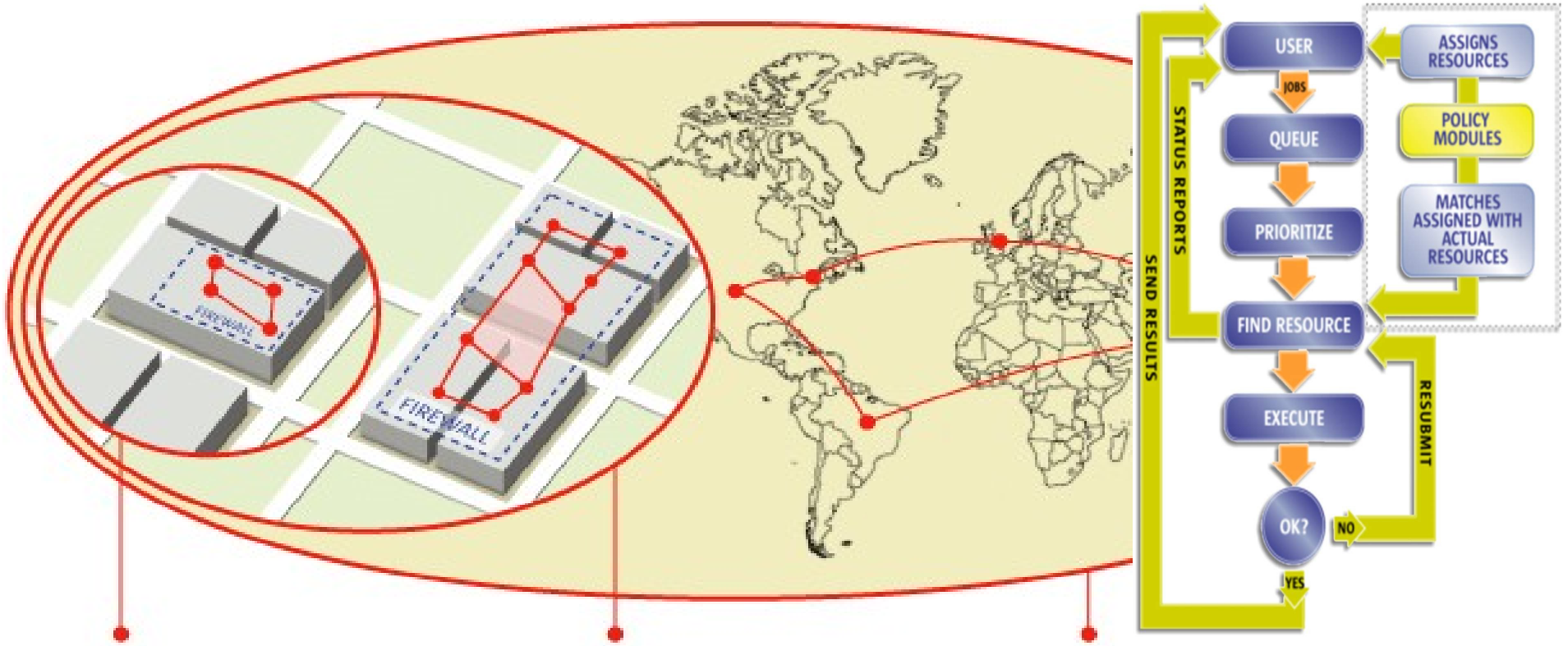
Future Directions...



Switch Gear / Network Only Offices



Secure Grid Computing



Cluster Grid

Resource Sharing

- Optimal utilization of resources
- Enable new capabilities/services
- Lower operational costs for IT
- Maximize productivity

Enterprise Grid

Java Web Services, Data Grid

- Policies ensure secure computing on demand
- Gives multiple groups seamless access to shared enterprise resources on demand
- Meets commercial enterprise standards

Global Grid

Trade Exchange

- Resources shared over the Internet
- Global view of distributed resources and data
- Trade Exchange for compute capacity and data access
- Foundation for Utility Computing

Security without Boundaries

Example: Multiple Companies and Sites

Always Connected:
Anywhere, Anytime, from Any Device

New York
Partner

London
Employee

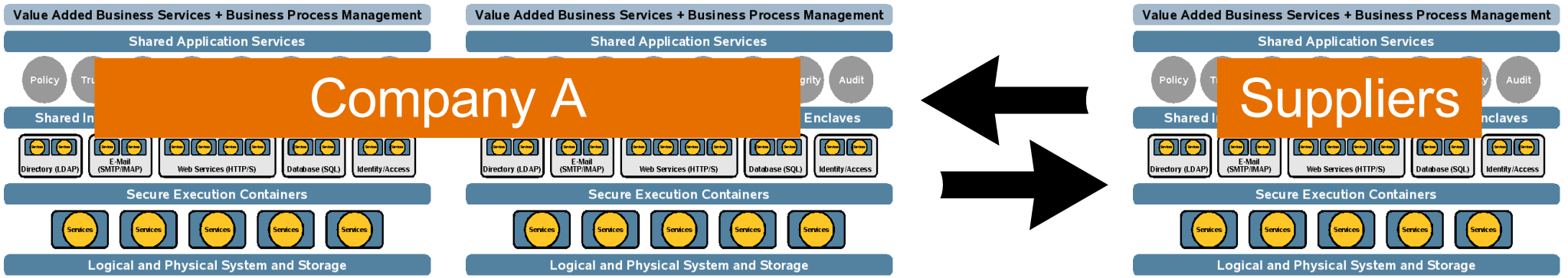
Chicago
Customer

Hong Kong
Supplier

Secure Desktop Services

Shared Presentation Services

Secure and Compliant Access
to Services and Data



Interconnected Enterprise Grids

Global Compute, Storage and Networking Utilities and Exchanges

Summary

1. Security building blocks and patterns offer a common way to view of security functions and controls.
2. Security patterns must complement traditional architectural, design and infrastructure patterns.
3. Sun Systemic Security provides the framework for understanding what patterns should be used when and how they should be instantiated.

For More Information

- Sun Security Home
  - http://www.sun.com/security
- Sun BluePrints Articles:
  - Toward Systemically Secure IT Architectures
    http://www.sun.com/blueprints/0206/819-5605.pdf
  - Sun's Pattern-based Design Framework: SDN
    http://www.sun.com/blueprints/0905/819-4148.pdf

Sun Systemic Security

Security Patterns for IT Architecture

Glenn Brunette

glenn.brunette@sun.com

http://blogs.sun.com/gbrunett/