Succeeding with SOA -- Recognizing Better Opportunities

Government and Industry SOA Symposium

April 21-22, 2010

Crystal City

Doug Tolbert, PhD
Distinguished Engineer, Federal Consulting Practice, CSC
dtolbert@csc.com

Everyone says SOA is great …

Long term total IT cost reductions up to 20% (Gartner)

Many non-cost benefits as well

- More flexible, better integrated IT architectures
- Mutually-independent, stateless services
- Speeds application development
- Promotes technology reuse
- Improves mutual alignment of business and IT concepts
- Encapsulates underlying technology platform details
- Automated service discovery
- Simplified testing
- Improved legacy application management

But if SOA is so great …

Why are there no more than a hand full of deployed SOA implementations in military environments?

- Even in the presence of clear DoD mandate to become “net-centric”
- And development of a number of relevant DoD specifications
  - NCES, GIG, and a number of others
- And some initial industry standardization efforts
- And a fair number of books
- And **lots** of product implementations

What do SOA deployment failures tell us that will increase the likelihood that future deployments will be successful?

Maybe this SOA stuff is harder than everyone says!

Some Reasons Why SOA Is Hard

Much of SOA is about aligning business process with technology process

- This is largely a sociological problem, not at technology one
  - And it may never get much easier!
- Often requires substantial effort and compromise by multiple stakeholders

Once the stakeholders have agreed …

- Abstracting services from existing system implementations is still hard
- Requires scarce talent that understands both technology and business issues

May require significant software (re)engineering efforts

- With all their well-known problems and challenges
- Some re-engineering projects may be trivial, but many important ones are not

SOA deployments are easier in some technology scenarios than in other

- Many are rather trivial
- But some really important ones are not
- Most of the time, must add SOA interfaces to existing systems

Our Scope Today

Ignore all that sociology stuff

- It's really hard, and a major benefit of SOA
- But we're talking about technology today

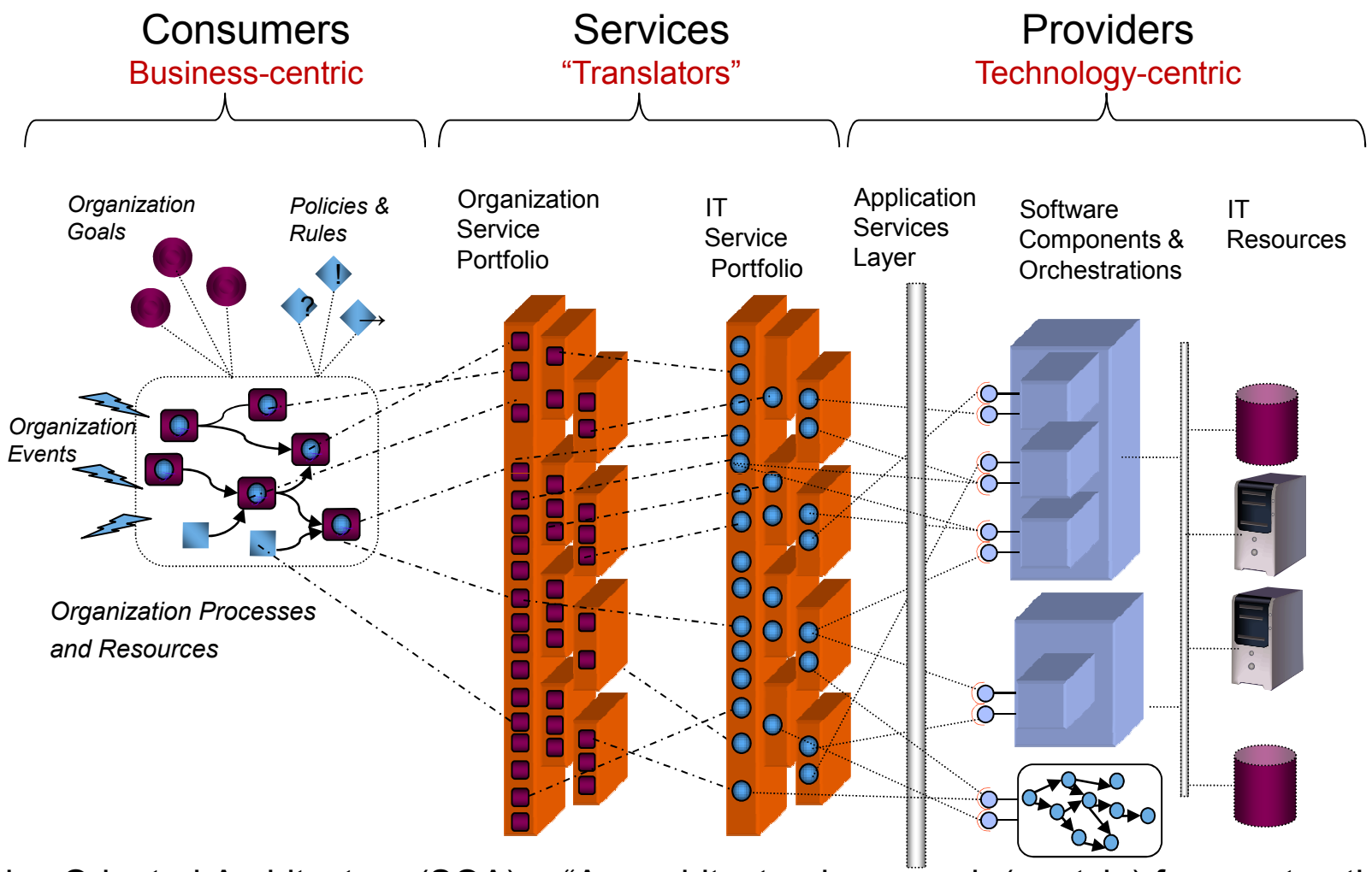
Focus on application technology scenarios

- As candidates for SOA deployments
- Some are good candidates
- Others are less good (= poor?)

Look for opportunities to make things better

- How might industry dynamics and trends help?
- Will standards light the way?

SOA Characteristics – A Reminder



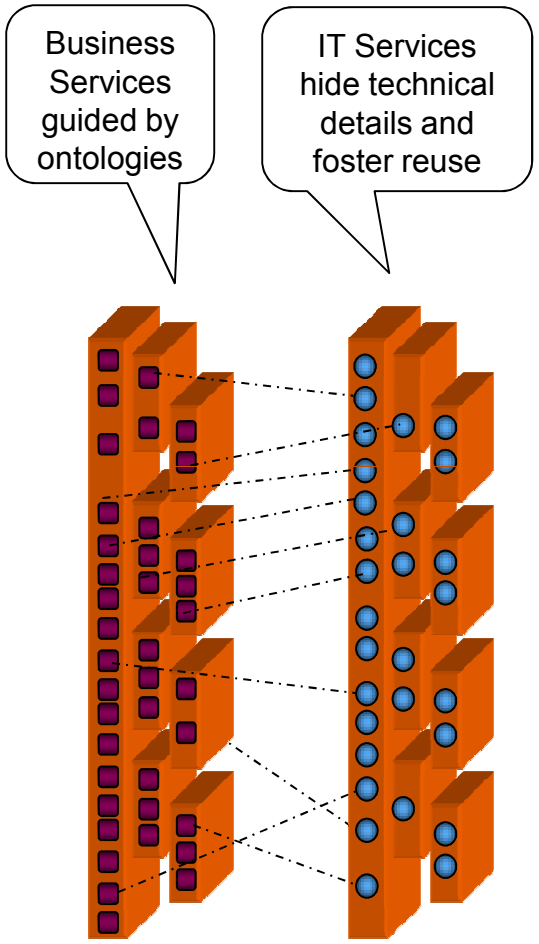
Service Oriented Architecture (SOA) -- "An architectural approach (or style) for constructing complex software-intensive systems from a set of universally interconnected and interdependent building blocks, called services" (Wikipedia)

Mature SOA implementations effectively define the "API" of an organization

SOA Deployment Scenarios – How to recognize “winners”

- Greenfield Business Services
  - Starting from scratch
  - Your programmers will love this one, but will you?
- Simple Modernizations
  - Inexpensive ways to start using SOA today
  - Legacy screen modernization
- Complex Modernizations
  - How SOA can help manage legacy migration and integration
- COTS systems
  - Understanding “componentization” opportunities
- ERP systems
  - Similar issues to COTS

Greenfield Business Services



New business service applications

- Start from scratch, no legacy implementation

Optimal situation from software design viewpoint

- Close match between business concepts and IT service interfaces improves communication between business and tech staffs
- May have to interface with pre-existing services and applications
- IT Services hide technical complexity from Business Services and foster reuse

Business services can be well-designed and separated from the beginning

- Develop shared stakeholder business ontologies
- Use them to guide design of business service interfaces

Greenfield opportunities exist, but are less frequent

- Almost everyone has a legacy to deal with!
- All new engineering – some cost/schedule risk

Simple Modernizations – Inexpensive ways to start using SOA

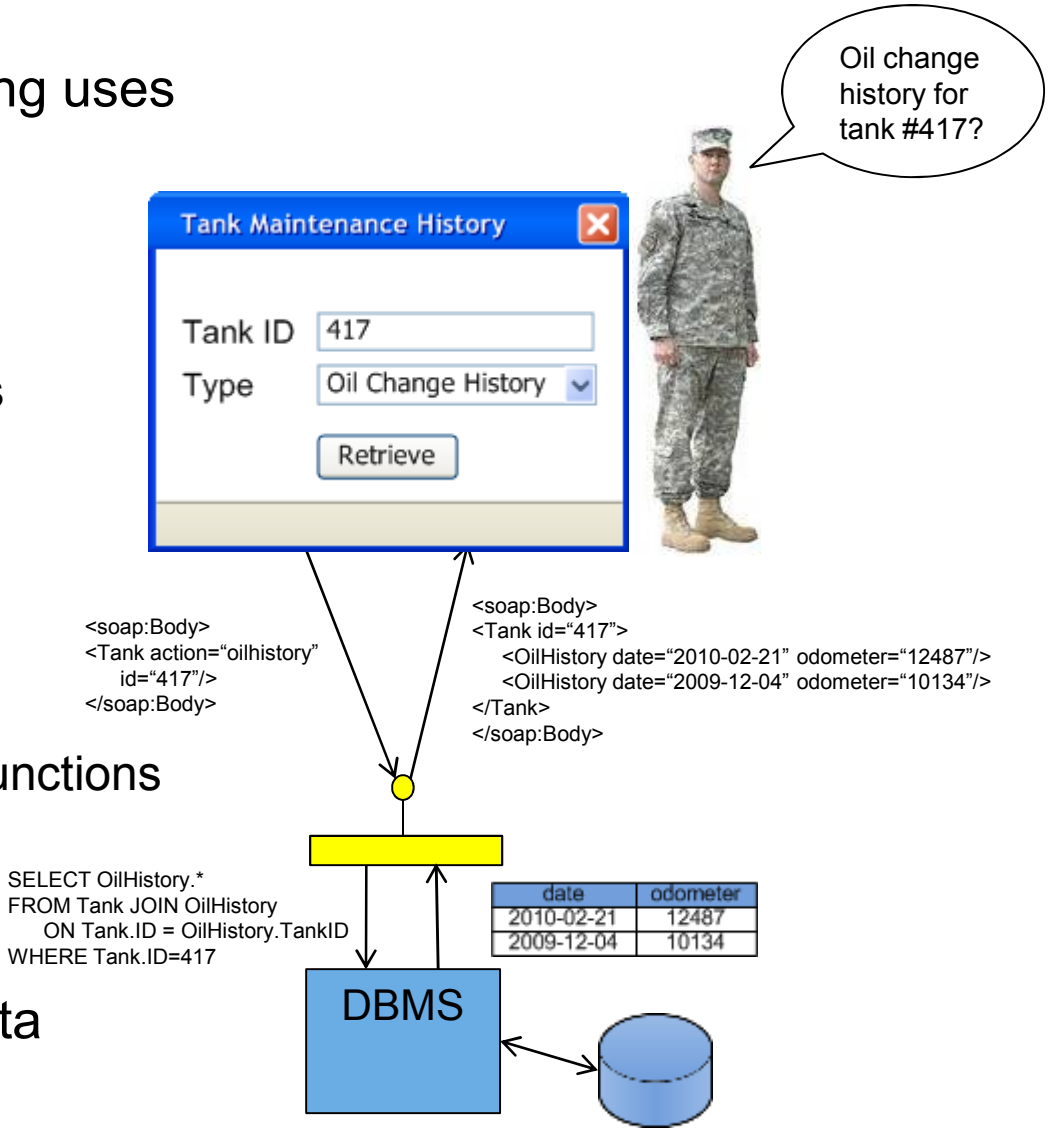
SOA interfaces for simple existing uses
- Data in existing databases
- CRUD wrappers
- Canned queries
- Deterministic business functions
- Limited record retrievals

Less good for
- Ad hoc queries
- Bulk, batch-like updates
- Dynamic or complex business functions

Simple to implement

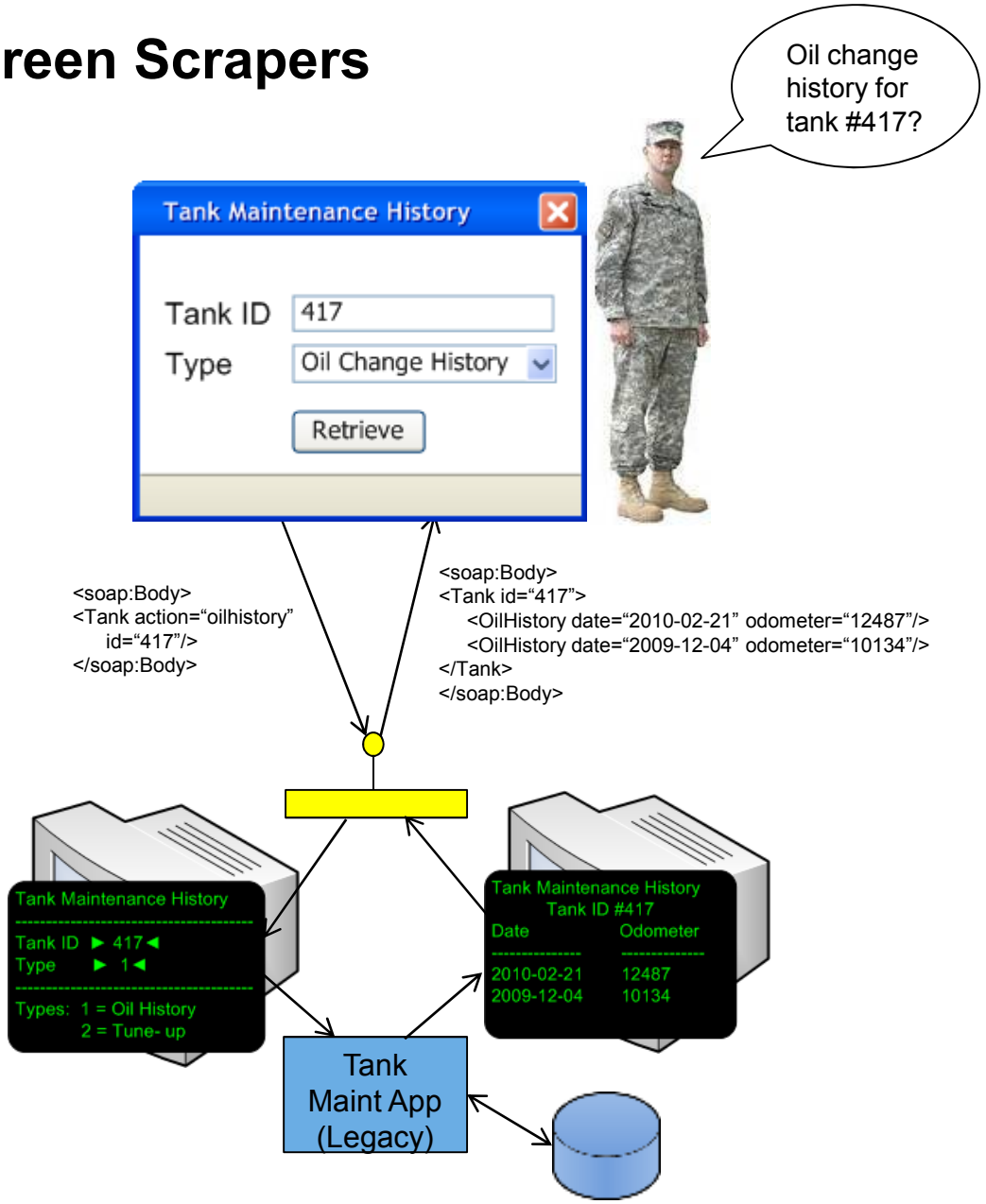
A “target rich” environment

Can greatly enhance level of data integration with minimal effort

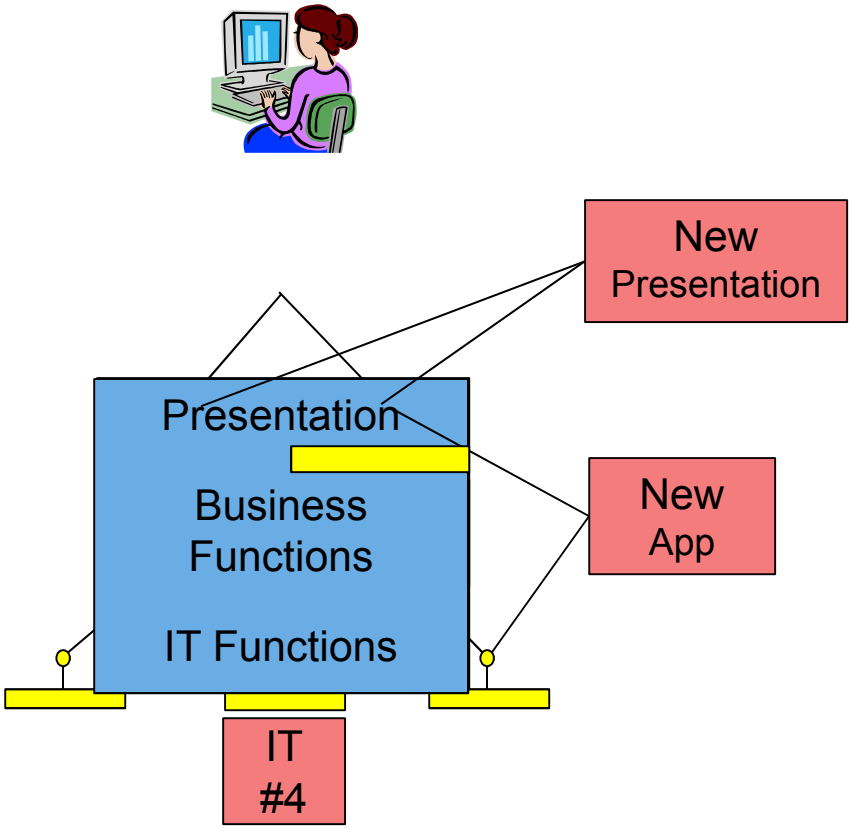


Simple Modernizations – Screen Scrapers

- Business functions presented thru legacy screen interfaces
  - DOS applications
  - Mainframe applications
- Creating SOA service enablers
  - SOA payload parameters and results formats are pre-deterimined by existing screen structures – less thinking required!
  - Service wrappers must mimic calls to underlying support software that servicing original legacy screen implementations
- Modernized in place without costly re-engineering
  - Often with NO re-engineering!
- Another target-rich environment
  - Screen scraper tooling available

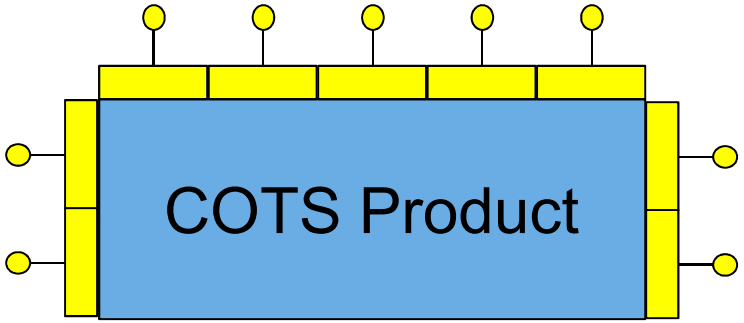


Complex Modernizations



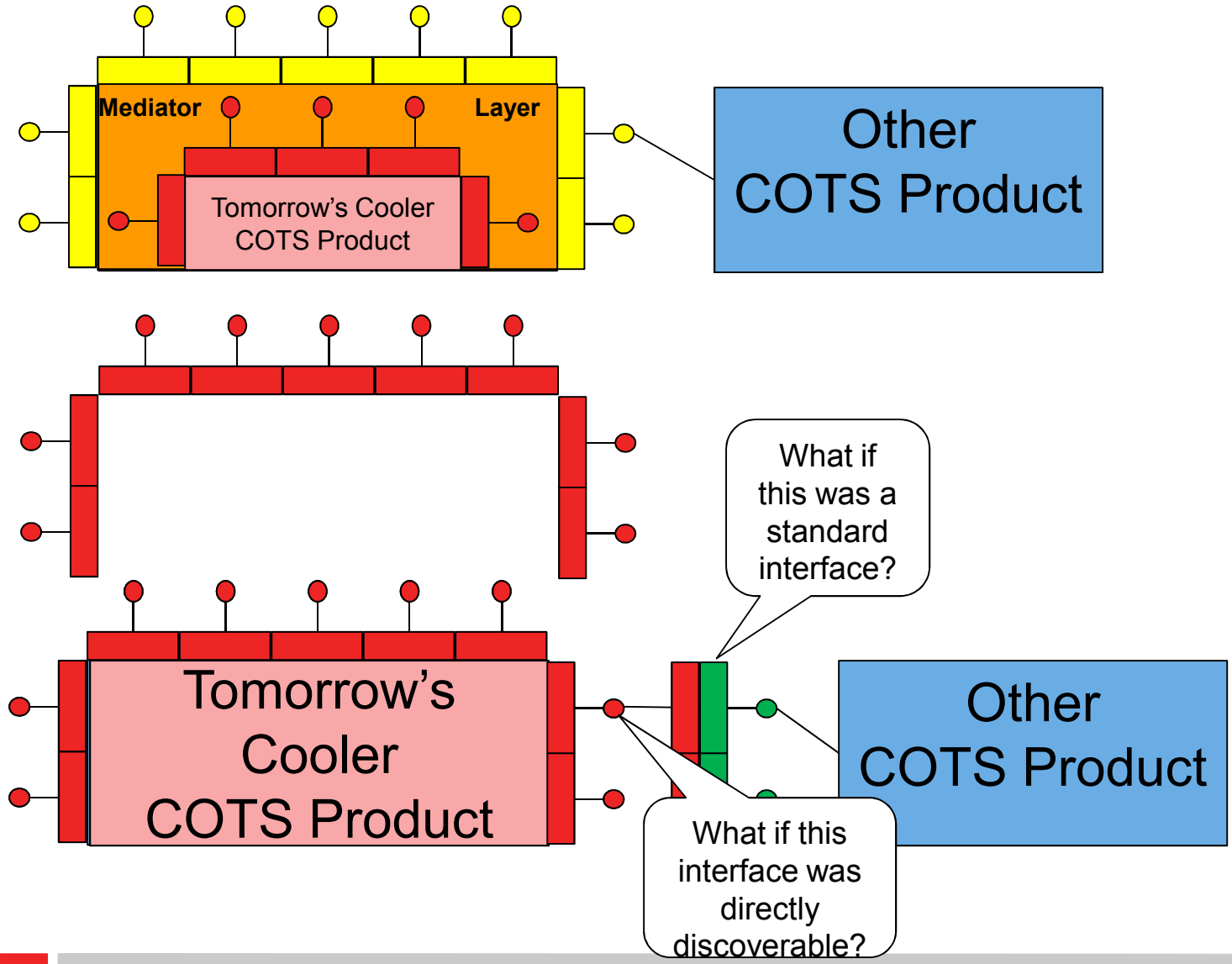
- Introduce business-level SOA wrappers to an existing legacy system to
  - Migrate technologies without service interruptions
  - Join modern application mash-ups
  - Become discoverable
- Refactoring monolithic legacy apps
  - Exposes embedded services to discovery
  - Promotes reuse in other applications
  - Allows new interfaces
- While giving owning organization some technology “breathing room”
  - Schedule and budget migrations at convenience
- Expect some re-engineering!
  - Cleanly separate business & IT functions

COTS Applications

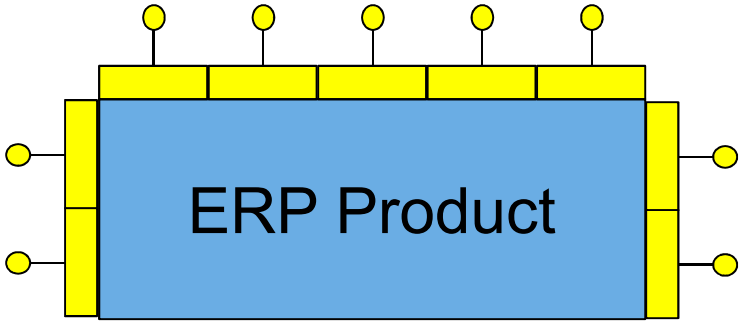


- COTS products are an efficient, cost-effective, low-risk ways to acquire proven business functionality
- Today, many COTS products are SOA-enabled out-of-the-box
  - eg BPM suites & business rule engines
  - Vendors may assume interoperability testing burdens for other well-known products
- But frequently, we must live with SOA interfaces provided by COTS software providers
  - Most are proprietary, few standards exist
  - So, choose COTS products whose SOA interfaces best match your application needs
  - Some SOA payload tailoring capabilities are available in higher-end products

COTS Applications -- Componentization



ERP Systems



- Like COTS, ERP systems are attractive because they are efficient ways to get lots of proven functionality
- But architectures of most ERP systems do not lend them to easy
  - Componentization (much like COTS)
  - Re-engineering for (unanticipated) reuse
- Users cannot dissect ERP systems
  - Can't use our complex modernization techniques here
  - Code refactoring requires major re-engineering effort that only the vendor can do
  - Must wait for ERP vendors to do it for us
- Again, we must live with vendor-provided SOA enabler services
  - Some vendors are starting to dissect ERP systems into smaller components

How standards can help SOA implementations

Standardization of component interfaces would promote easier replacement of COTS components

- As newer, possibly more capable, components become available
- Vendors are not particularly motivated to help with this, though
- One reason Standards Development Organizations (SDOs) exist

Standard descriptions of services

- Promote discovery of previously-unknown service providers with little or no development
- Could service consumers automatically reconfigure to match service provider characteristics?

What technologies will be necessary to support such a capabilities?

- Model-driven technologies will be essential
- Agent-based negotiators and complex event processing will likely be important
  - Especially in more complicated service scenarios

CSC Standards Activities Target SOA

- CSC actively participates in industry standards efforts
  - At a level unusual for a systems integrator
- Object Management Group
  - SoaML leaders
  - Active in related technologies: UML, MDA, CWM, SysML
  - Agents & event technologies: negotiable discovery of SOA interfaces?
  - Leading definition of fledgling cloud standardization roadmap
- Web standards
  - SOA Reference Model
- These are the kinds of technology standards that will help government and industry organizations achieve the substantial promise of the SOA architectures and related technologies

Summary

- SOA may not be as easily deployed as some would have us believe
  - Understand the characteristics of underlying technology environments
  - Don't underestimate the cost of system refactoring needed to achieve SOA
- Nevertheless, there are a number of good opportunities
  - Greenfield applications – do it right the first time!
  - Many simple modernizations
  - Value-justified complex legacy modernizations
  - Feature-rich vendor COTS and ERP systems
    - Vendor has done most of the work for you already
    - But there may be some flexibility constraints
    - Evolve with your vendor as they evolve
- Standards community is developing some promising techniques
  - Enhanced, ontology-based service discovery seems very feasible
  - Model-driven service consumer reconfiguration to match service providers?