CREDIT SUISSE

13 Years of Service Oriented Architecture@Credit Suisse: Lessons Learned – Remaining Challenges

Keynote ECOWS 2011 – Lugano, Switzerland
Date: 16.9.2011

Stephan Murer, Chief Architect, Credit Suisse

Credit Suisse IT Division – Key Facts

Large and complex environment

- **Significant technology footprint** and **high degree of complexity**
  - 67k users supported in 550 locations, 4 major data centers plus failovers, 95k workstations / laptops
  - 17‘100 internal & external employees
  - Complex processing environment (>6'000 applications) evolved over many years
- **Considerable IT spending** – CHF 3.7bn 2011 budget, with a growth rate of 8% over the last 5 years
  - CHF 1.8bn investment into new functionality in 3'000 projects

Ongoing commitment to efficiency

- Continuous **focus on sustainable IT unit cost reduction**
  - We already deliver some of the **best infrastructure unit cost efficiency** in the industry (e.g., servers, mainframe, storage)
  - Delivered 20% **reduction in blended workforce rates** over the last four years
- **Contained maintenance cost growth** over the last few years, maintaining our ability to invest in new capabilities and led to a leading maintenance / investment ratio

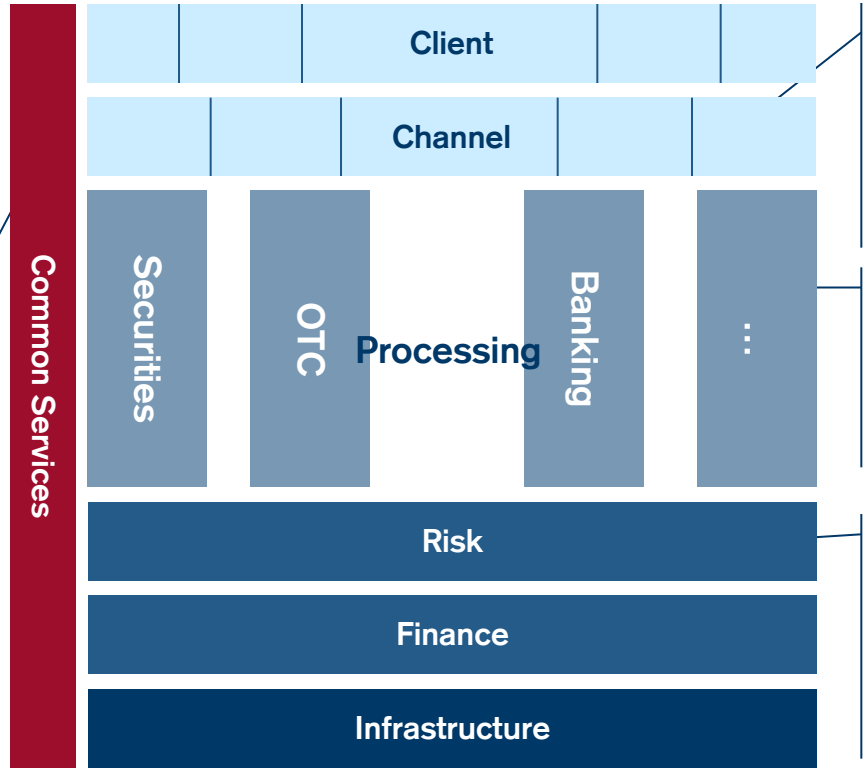
IT as catalyst for the bank's growth

- **IT delivery** of projects and services is **consistent** and **meets businesses expectations**
  - User satisfaction generally increased over past few years, although 2010 partly saw a decline
  - IT production services increased availability significantly through globally aligned and regionally executed stability programs
- IT continuously seeks to **enhance** the bank's **operating efficiency**
- **Industry recognition** for some of our **innovative IT solutions**, e.g., Best treasury/cash management (European Banking Technology Awards); best use of IT in Wholesale & IB, HALO – best trading platform, cash flow & liquidity – best compliance project (Financial Sector Tech Award)

Convergence to more agile and standardized operating model to improve our capabilities

Shared components

- Single sources of reference data
- Common SOA
- Common domain and business services model



Consistent client experience

- single sign-on
- one portal, multiple solutions

Product processing

- shared, legal entity-agnostic global processing platforms

Risk mgmt and financial control

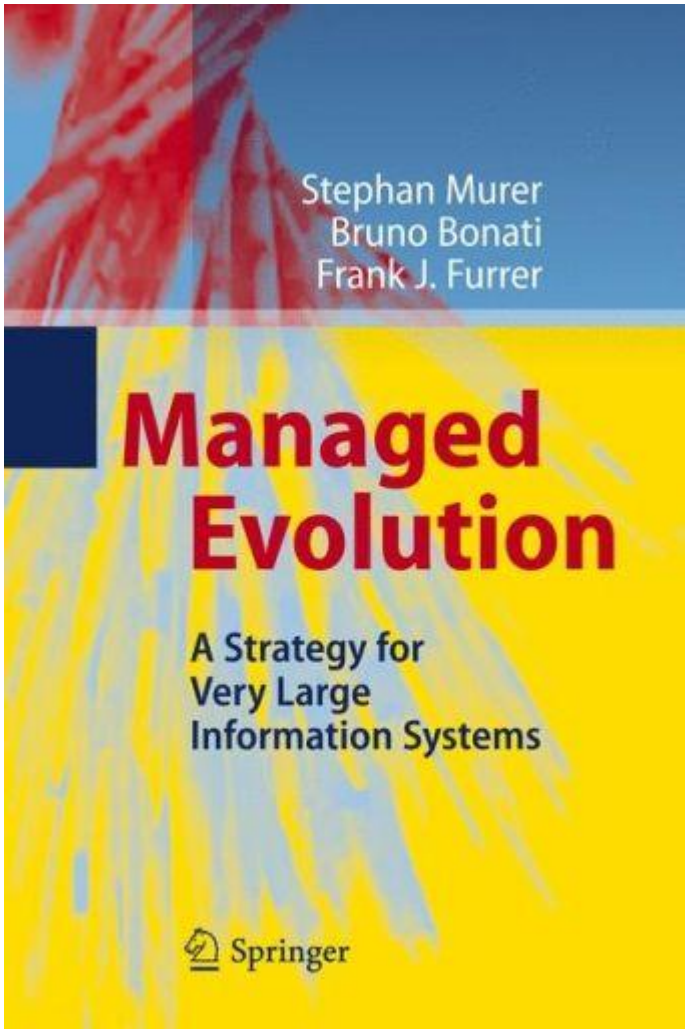
- unified risk and finance platforms, based on agreed front-office data

Common infrastructure

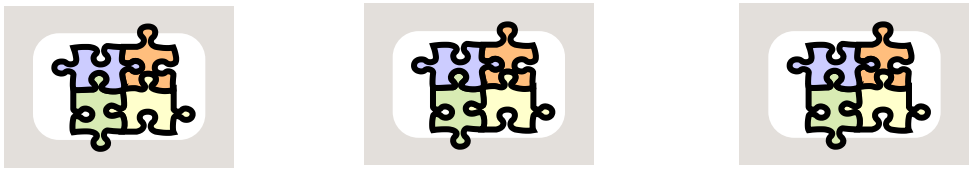
- Common infrastructure based on consistent architectural standards that support our systems in a scalable, agile, and cost efficient manner

Managed evolution of very large systems requires stable interfaces

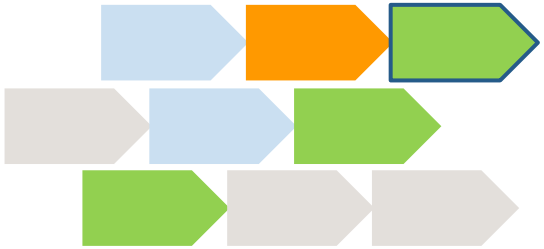
- Hard to replace a very large information system as a whole
  - High cost (>1bn CHF, estimate for Swiss platform)
  - Development time too high (>5years)
  - High risk, as both technical and business prerequisites shift over time
- Managed evolution is the only feasible approach
  - Stepwise transformation of landscape, renewing component after component
  - Multi-year effort which gradually implements the target architecture
- Well encapsulated components exposing managed interfaces are a prerequisite for managed evolution:
  - Technically renew components, without affecting clients
  - Consolidate redundant data and functionality behind common interfaces and simplify
  - Seamlessly operate a technically heterogeneous system resulting from the evolutionary approach



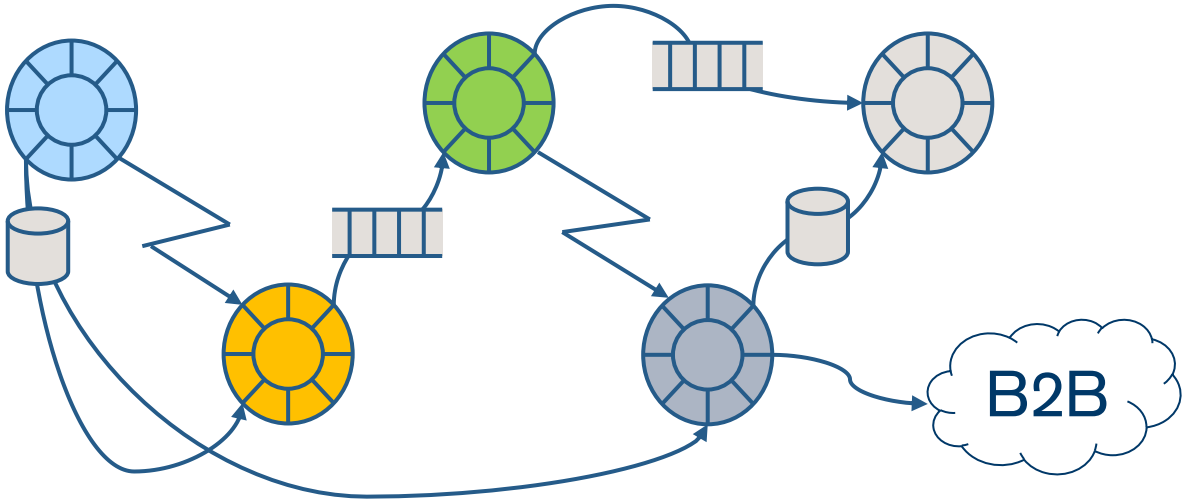
Comprehensive Integration Architecture



- UI Integration, arranging desktop real estate, managing UI context
- Core standards: HTML5, Java portlets

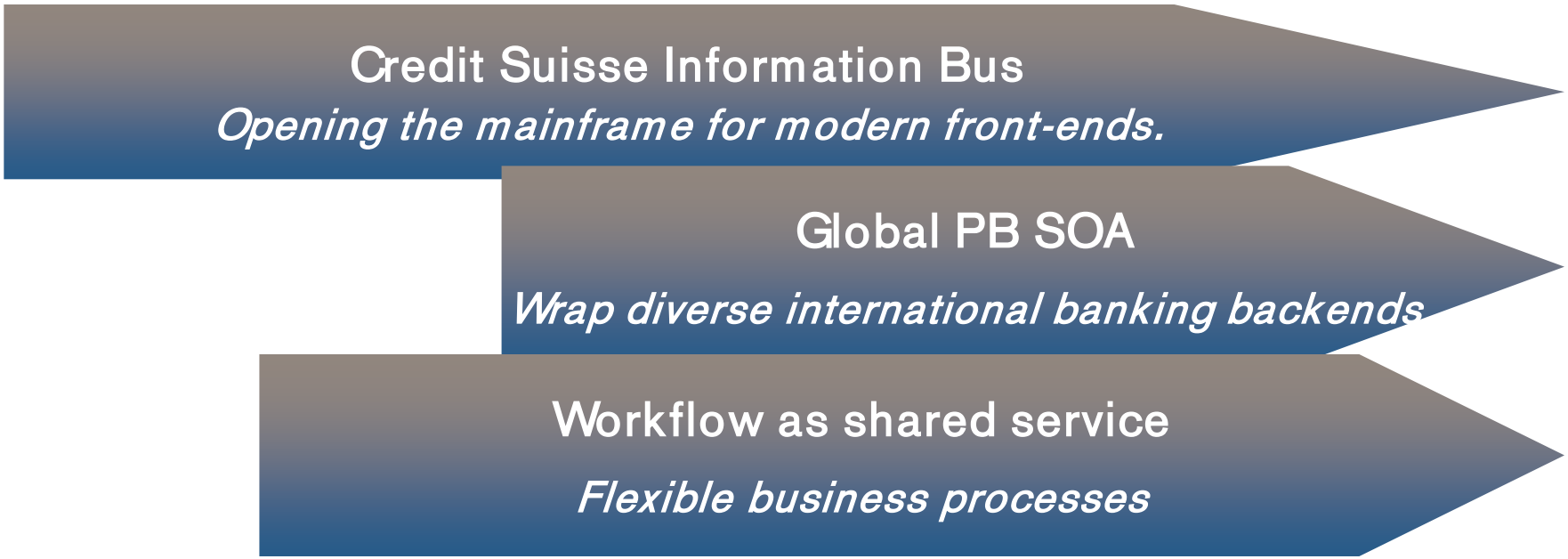


- Process integration, managing process context
- Exposing UI components for process control (task list …)
- Core standards: BPEL



- Function and data integration (core SOA)
- Based on managed interface contracts
- Exposing UI components and functions for process composition
- Core standards: XML, WSDL, IDL, FTPS, SOAP, IIOP, MQ, domain standards (ISO 20022, SWIFT, FPML, FIX)

SOA Experience @ Credit Suisse
3 case studies



Case 1: Credit Suisse Information Bus (CSIB)

Facts

- Started in 1998, more than 1000 services built up to now
- All applications on the Swiss Platform offer and/or consume services today

Objectives

- Enable Managed Evolution of Mainframe Platform
- Component architecture for the Swiss Platform
- Reuse of core data&functionality (mainly) residing on the mainframe for modern front-ends

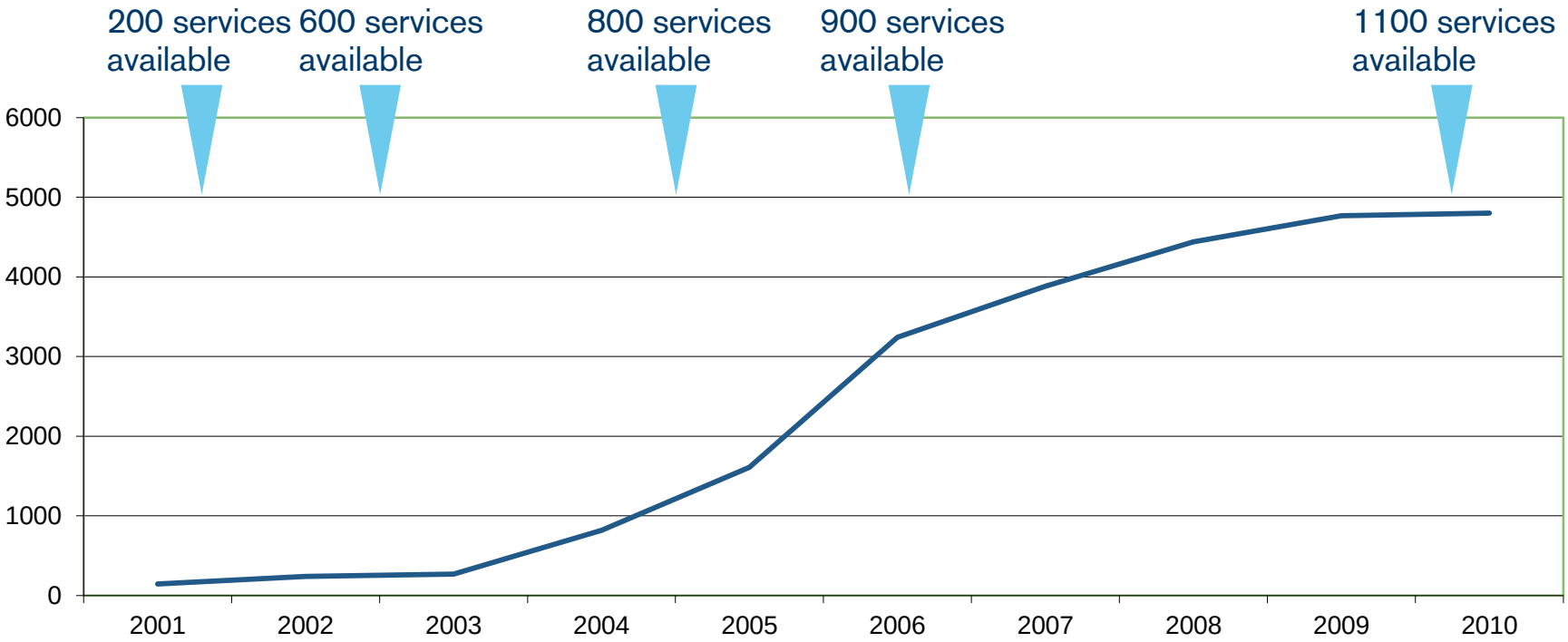
Technology

- Orbix - CORBA for synchronous services
- WebSphere MessageBroker / MQ for messaging
- Connect:Direct for files

Footprint

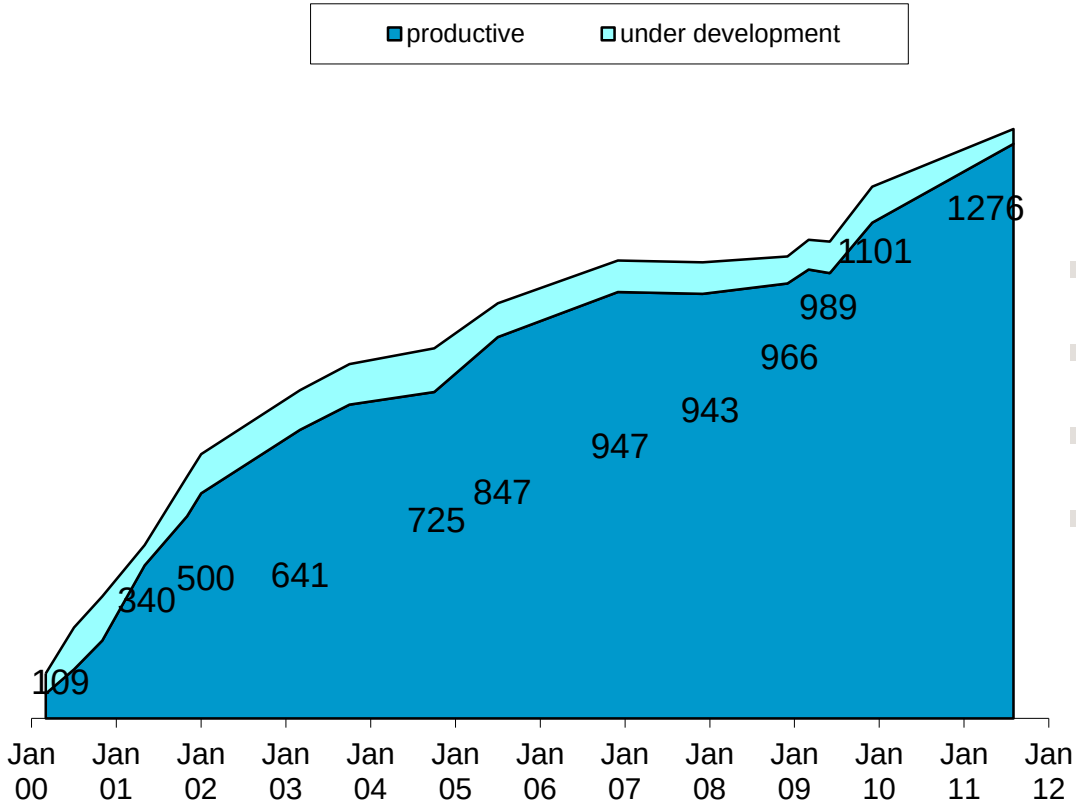
- About 1000 public CORBA services, 70 message publishers
- 400 Mio. CORBA calls & 120 Mio. messages delivered per month

Measure Progress
Use of services follows availability



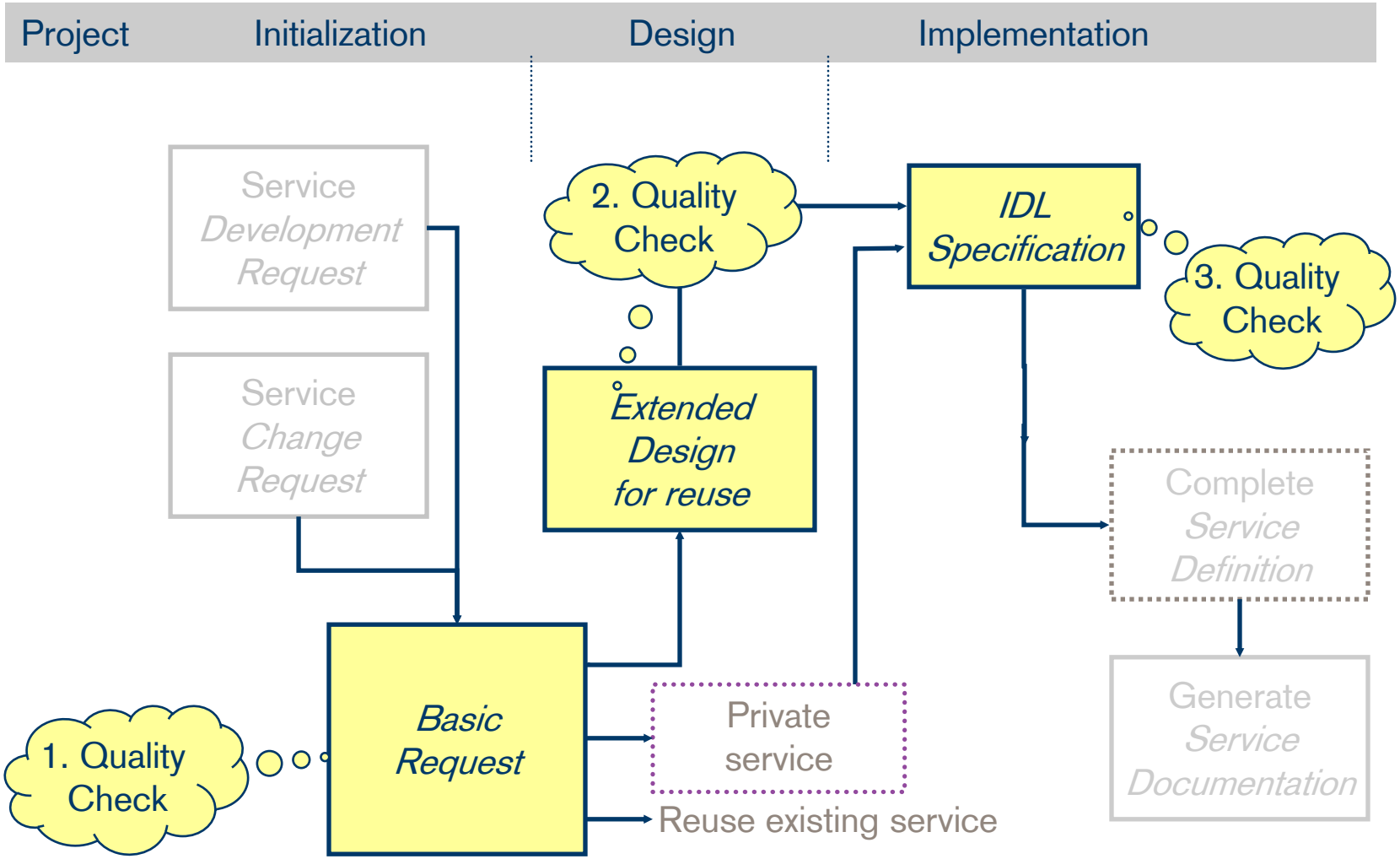
- Wide use of services follows a critical mass of available services
- Today ~ 5 billion service calls a year in Switzerland
- Fully decoupled platform some years ahead
- Core banking system on mainframe completely decoupled from remainder of platform

Measure Progress - Services built as needed

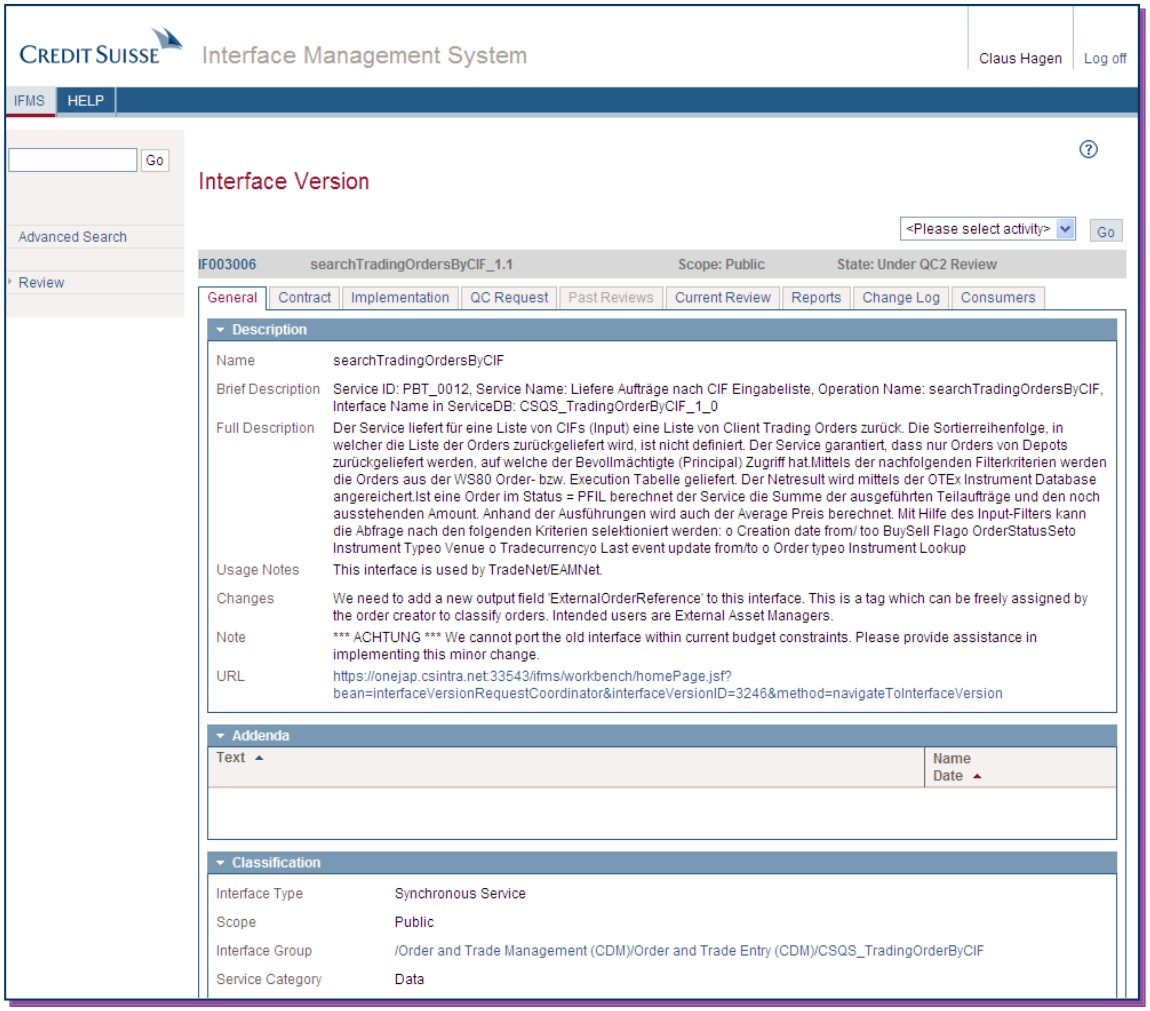


- Build services as they are needed
- Gradually decouple domains
- Use existing functionalities
- Bulk services from 2009

Governance – bottom-up requests, top down QA



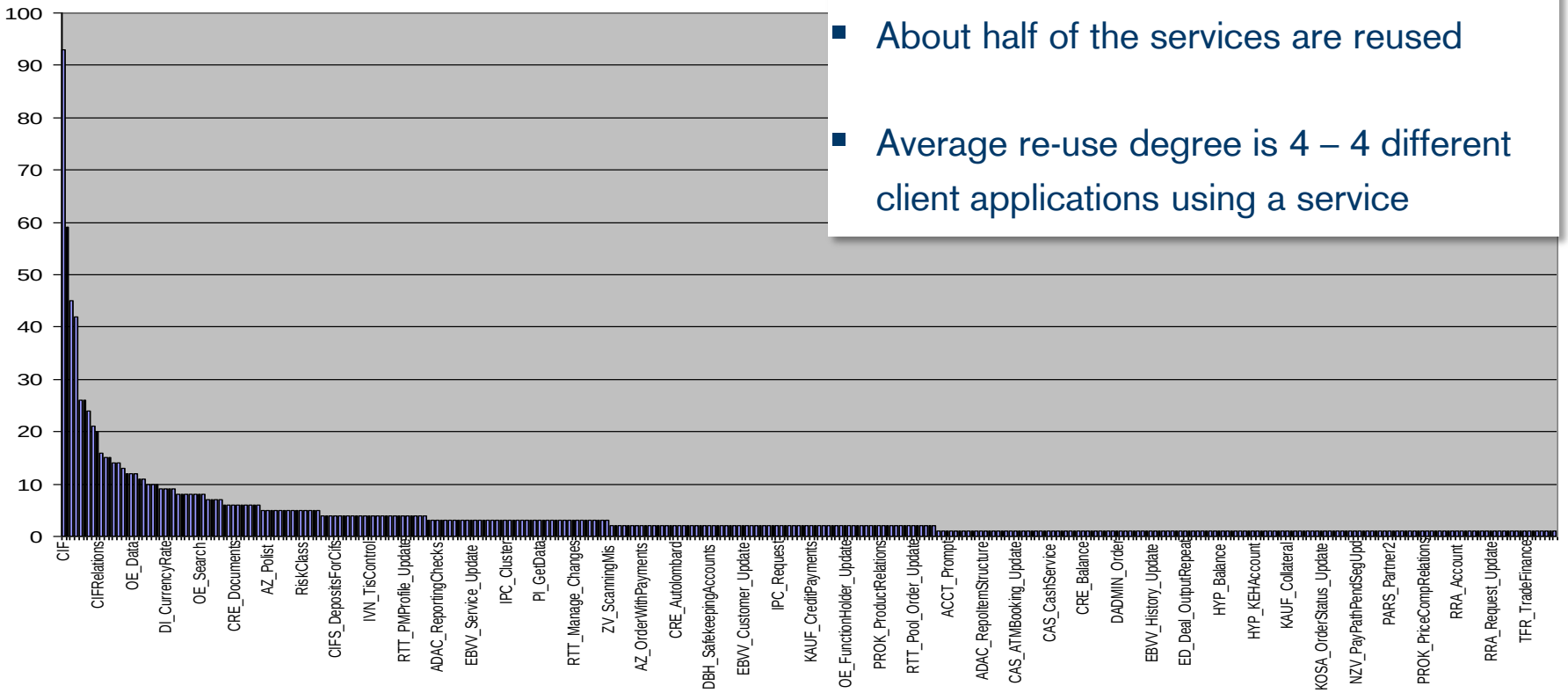
Governance - Interface Management System



- Searchable Service Catalog
- Data type repository with UML/WSDL integration
- Governance workflows
- Lifecycle management
- Extensible code generator (MDA)
- Integration with other tools, like application repository or accounting

Measure Reuse
Reuse is very uneven

- Re-use of services varies based on type of service. Reference Data Services have the highest re-use.
- About half of the services are reused
- Average re-use degree is 4 – 4 different client applications using a service

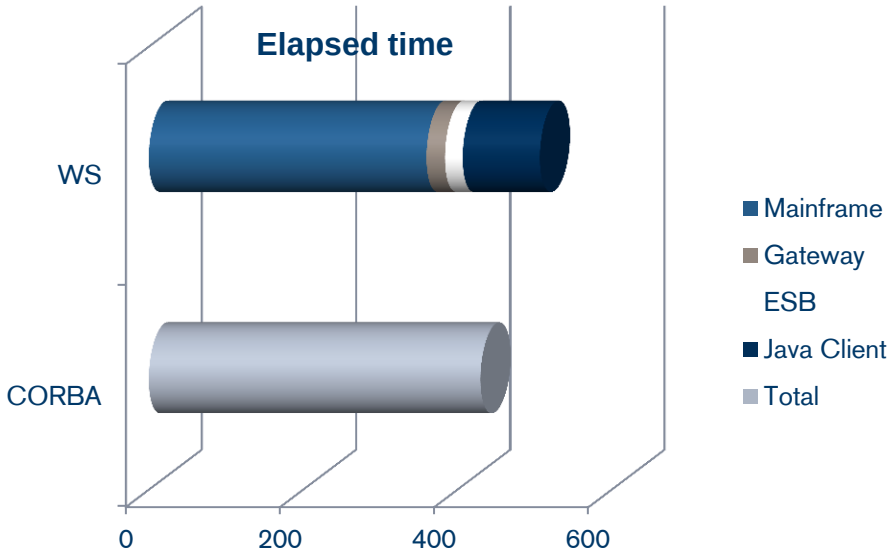


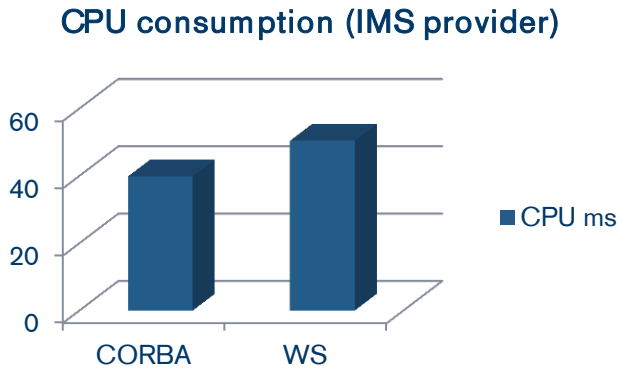
Technology Migration CORBA -> WebServices

- Initial performance of new middleware will hardly match the existing, mature infrastructure
- Thorough performance analysis with gradual improvements on all layers of the stack is required
- More verbose protocols (XML, SOAP) harder to implement efficiently

Example:

- Initial results of web service performance against CORBA
- Mainframe as server, Java Client
- Service retrieves details for 900 accounts
- Elapsed time in ms





CS Information Bus - experiences

Strengths

- Well-accepted, high proliferation, good re-use of services
- Solid middleware foundation
- Governance: Combination of bottom-up approach and top-down QA works

Challenges

- Management becomes difficult because of size (+1000 services)
- Business object model (BOM) required as a framework in which to manage the service landscape (currently in development)
- Long term migration towards Web Services, without compromising performance

Lessons learned

- Strong governance is essential
- Organization needs time to learn
- Bottom-up approach helped in gaining critical mass, but...
- ...supporting service design with common data model would greatly improve service quality and simplify QA
- Need improved interface management system (IFMS)
- Formal service descriptions could be the basis for code generation

Case 2: Global Private Banking SOA

Facts

- Started in 2005
- About 200 Services built up to now
- Services implemented in different countries (Monaco, UK, Germany, Singapore)

Objectives

- Re-use the same frontend applications with different local backends
- Initial driver: Standardized front desk application
- Currently we see a wave of new frontend applications

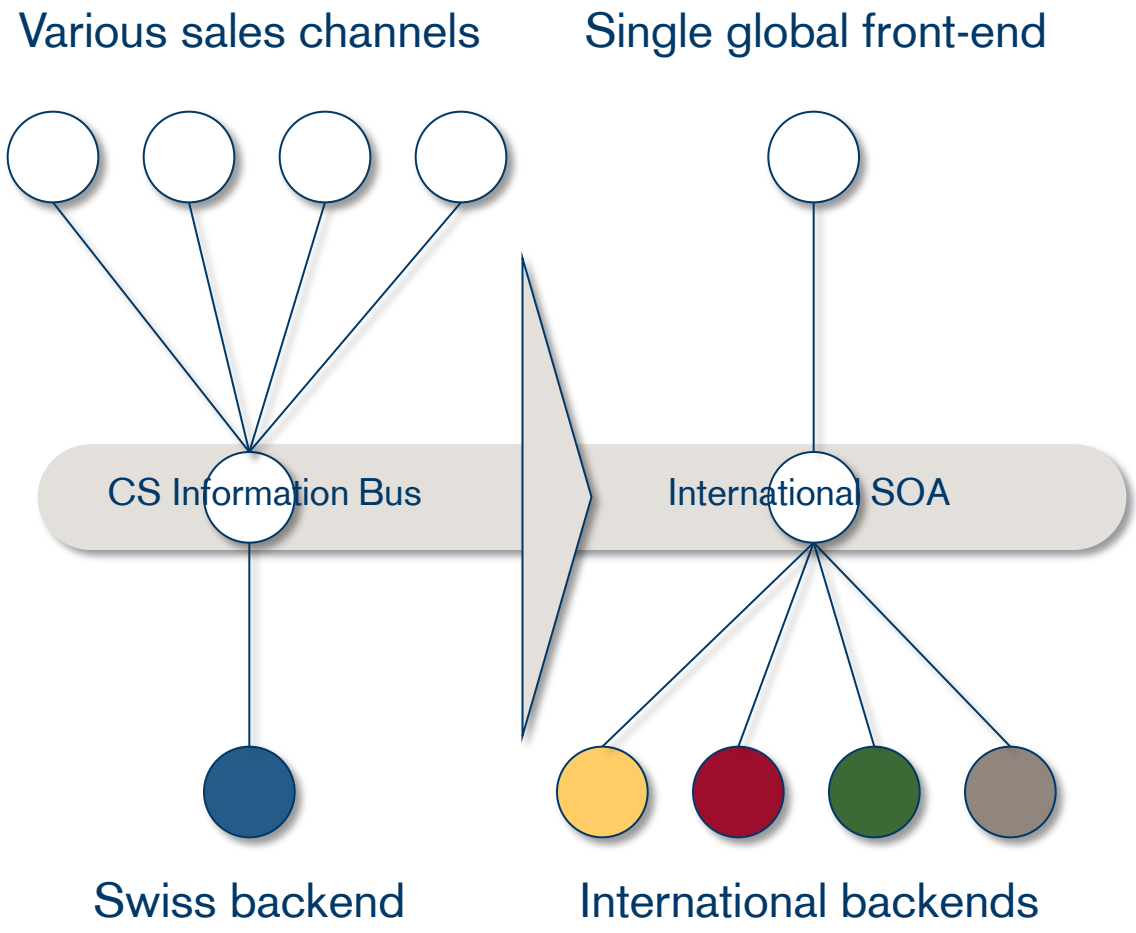
Technology

- Web Services (only synchronous communication needed)

Footprint

- Comparatively small number of interactions today.
- Large growth due to implementation of global strategy

From many clients using the same service to many providers offering the same service



Service provider defines semantics

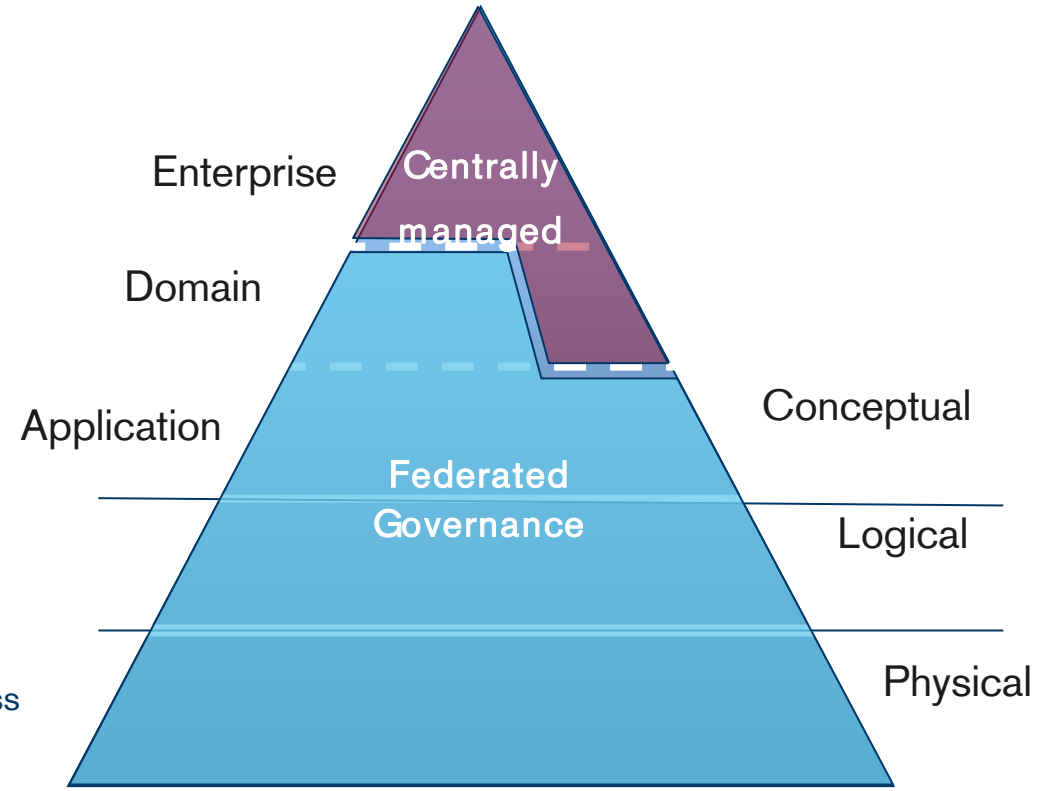
Implicitly defined with single service provider

Challenging with multiple service providers

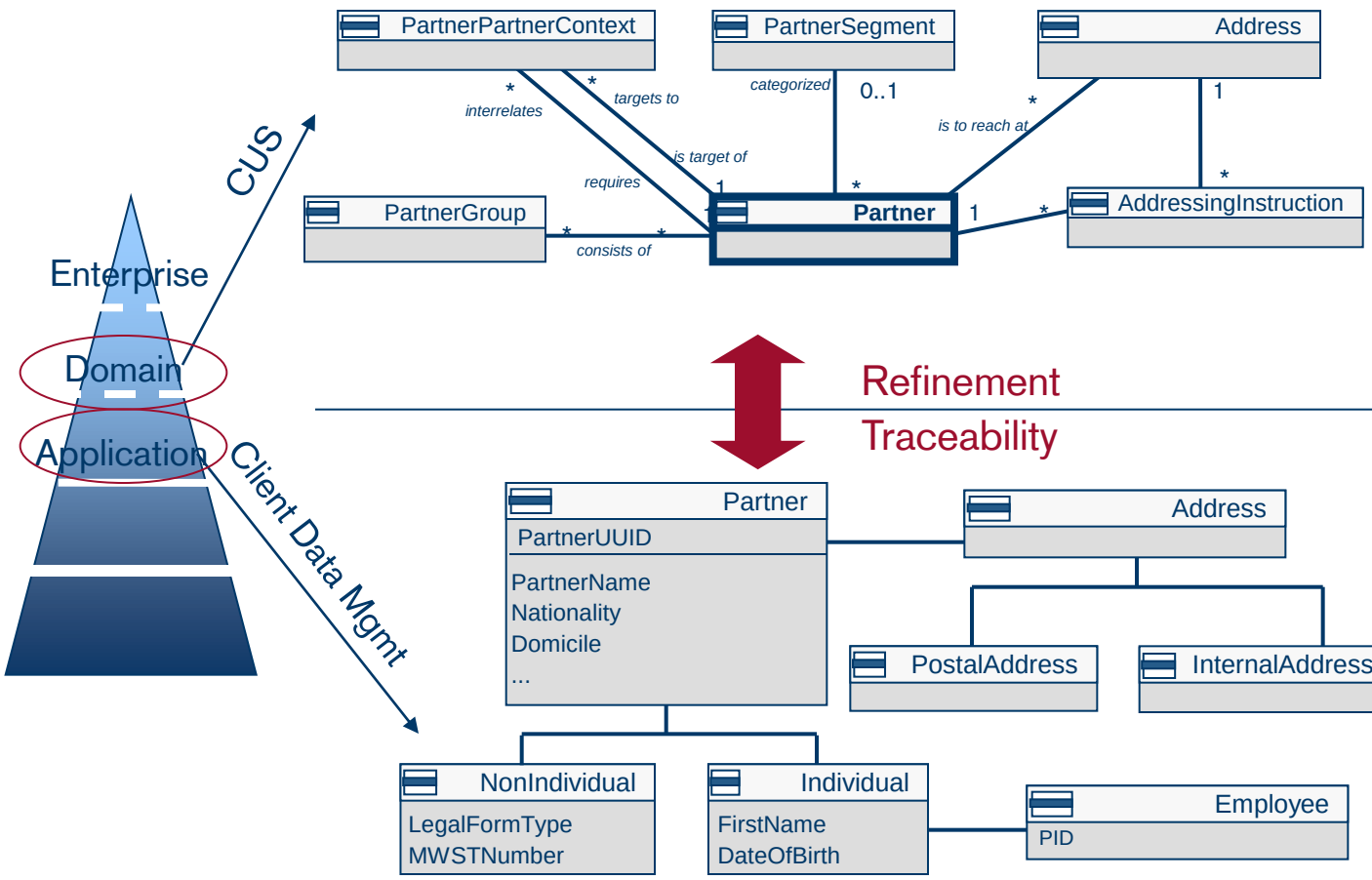
Explicit semantics, especially explicit information model needed.

Business Object Model – Common model semantically aligns information

- We need to define the precise semantics of data exchanged through services
  - SOA means many producers communicating with many consumers
  - Bottom-up service design needs overarching information model to ensure semantic consistency
  - Top-down approaches to service design require information model defining the overall structure
- Business Object Model serves these purposes:
  - Provides reference naming data elements
  - Typed data elements for correct representation
  - Relationships and annotations give semantic clarity
  - Model is adopted on a conceptual level as a basis for defining service data
  - Model used as a basis for implementing business objects
  - Establishing awareness of model as an abstraction of the requirements variations across locations



Business Object Model – Linking heterogeneous implementations to common semantics



- Owned by domain – with all the responsibilities
- Refinement and traceability
- High-level ownership

- Scope of a single application
- Foundation for logical / physical design
- Stewardship (like ITOwner for apps)

Global Private Banking SOA - experiences

Strengths

- Re-uses proven governance developed with the CSIB
- Adds logical data model to improve transparency and quality

Challenges

- Dealing with the heterogeneity of different locations and their local backends
- Dealing with a large diversity of teams and responsibility in the different locations
- Long-term blur of distinction between Swiss and International platforms

Lessons learned

- The business object model really helps. Need to introduce it for all SOA environments
- We are still learning how to balance local agility needs and global coordination/quality requirements

Case 3: Workflow infrastructure – Process Integration

Facts

- Started in 2000
- Currently broadly used internationally

Objectives

- Decouple process management from application logic
- Flexible service orchestration to support adaptation of application landscape to different business processes

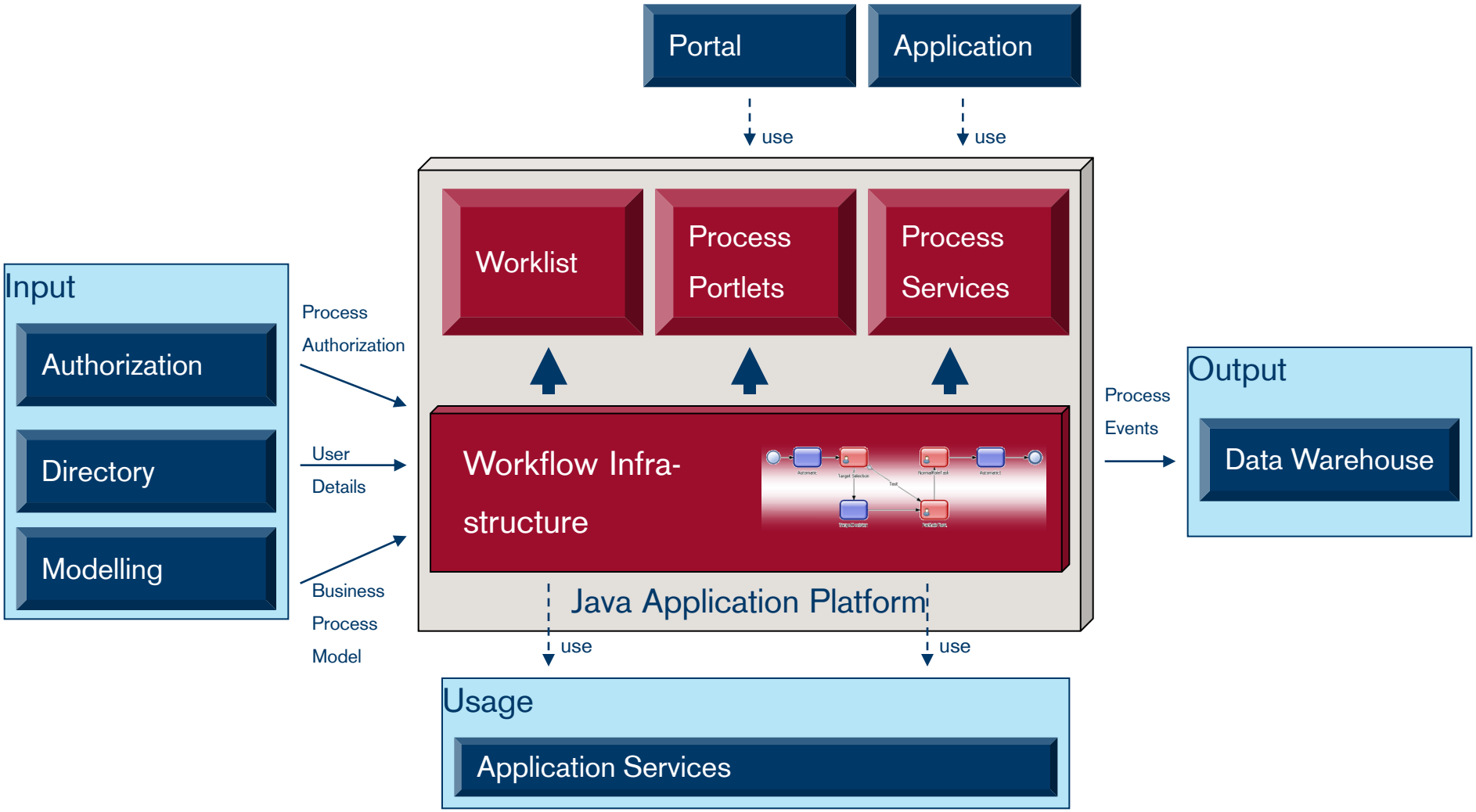
Technology

- Oracle BPM (current technology), IBM MQSeries Workflow (past technology), based on BPMN

Footprint

- In use globally in more than 25 Applications, with about 150 processes types. About 100'000 process instances per month.

Workflow Infrastructure – Architecture embedding



Employee Onboarding – Setting up IT Accounts

Hire new employee

Challenge:

New hires could wait several days to begin work until they have access to all systems

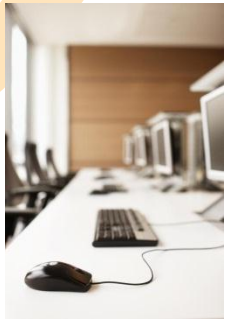
Issue smartcard

Create HR record

Create email account

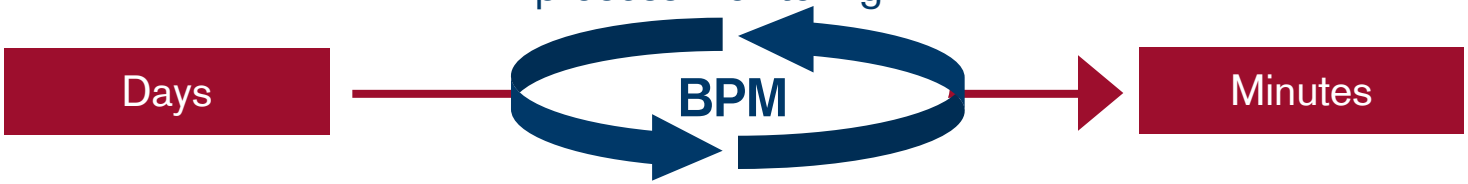
Create user identity

Set up IT accounts

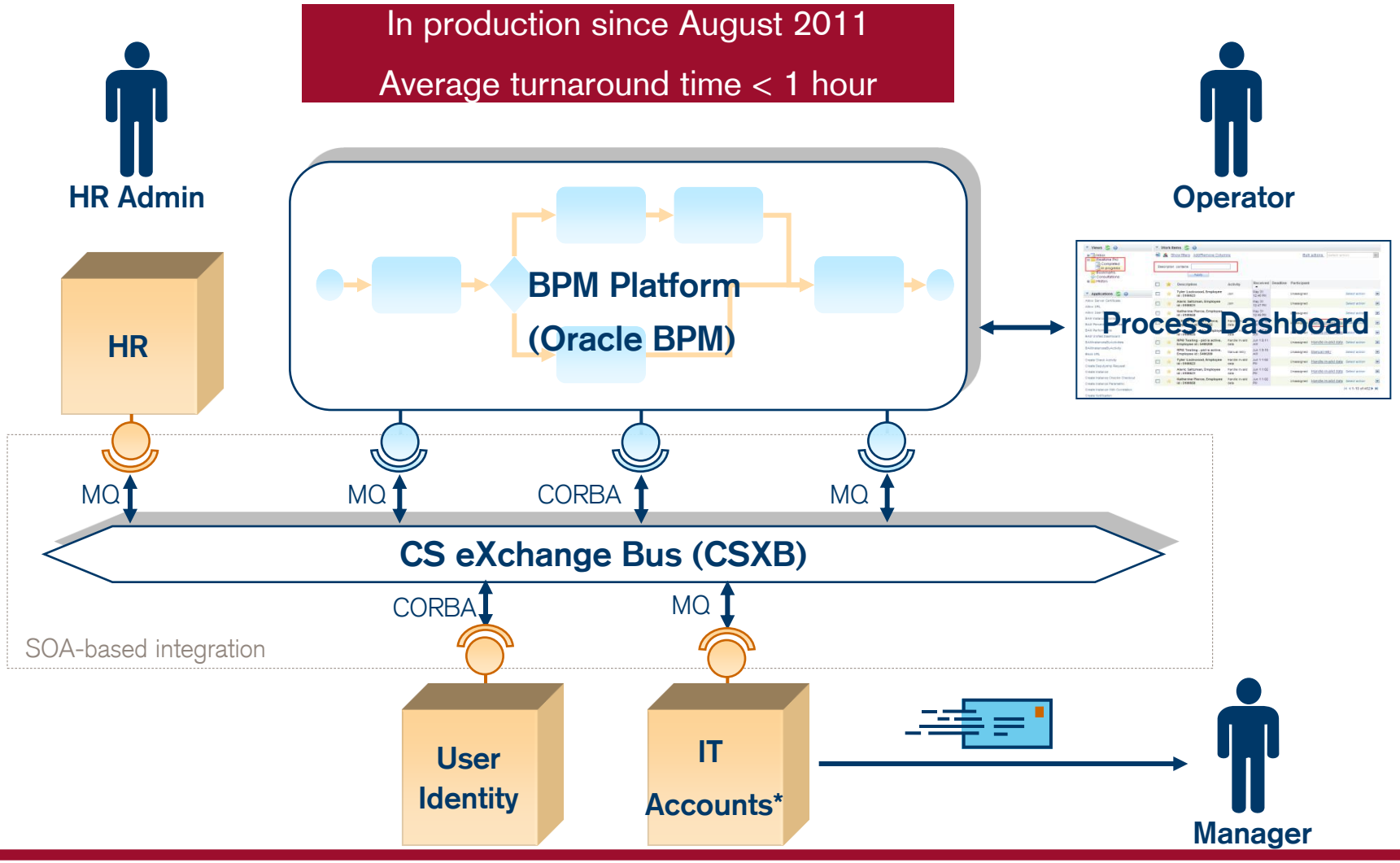


Process automation - from days to minutes

Pain Points	Workflow Solution	Benefits
• Several days to set up IT accounts for new employees	• Workflow Platform automates IT accounts creation in "near realtime"	• Improvement of overall employee onboarding experience
• "Quick hires" delayed in being productive	• HR events used to trigger account setup	• Immediate productivity for "quick hires"
• Different processes for different regions	• Process steps are orchestrated as SOA services	• Reduced complexity
• Complex processes and lengthy trouble shooting	• Core process steps unified globally	• Unified processes across regions
• No reliable metrics for process improvement	• Metrics gathered by process monitoring	• Continuous process improvement enabled by gathered metrics



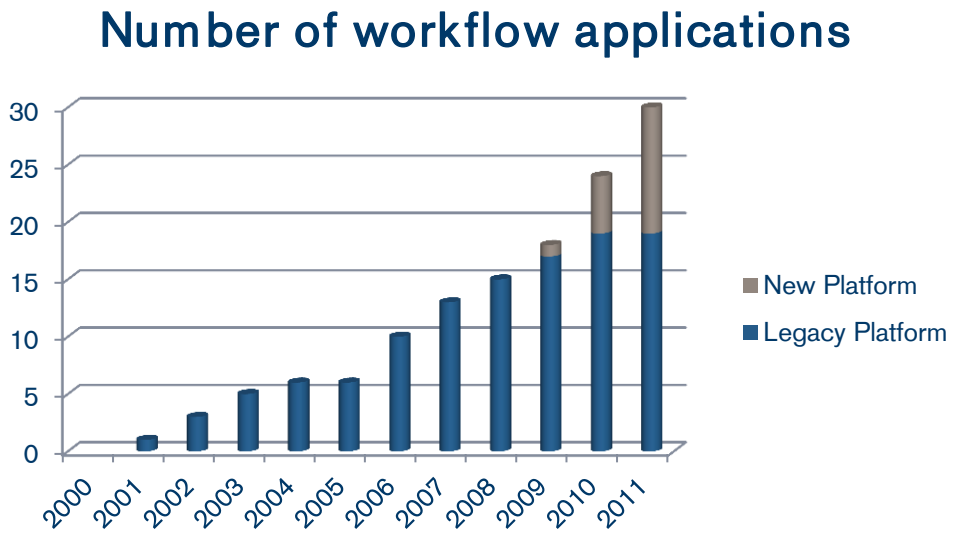
Service orchestration through managed interfaces



*IT account provisioning is handled by several different systems, which are not shown here in detail.

Workflow as shared infrastructure - KPIs

- Shared infrastructure used by multiple applications
- Central engineering, support and consulting team
- Infrastructure available in all major hubs
- Since 2010 transition from legacy solution (IBM MQSeries workflow) to new technology (Oracle BPM)
- Decouple workflow apps from server product as much as possible



Key numbers (Aug 2011)	Legacy platform	New platform	Total
Applications	18	8	26
Process models	118	28	146
Process activities	748	159	907
Process instances p.month	109'000	48'00	114'000

Workflow Infrastructure - experiences

Strengths

- Widely accepted across the firm
- Clean isolation of workflow engine through robust APIs proved successful. Engine is being exhanged without too much impact on applications.

Challenges

- Often off-the-shelf applications come with built-in workflow capability and don't lend themselves for external orchestration.
- Competition from domain specific orchestration tools, leading to early successes, but failing in enterprise-robustness and end-to-end orchestration.
- Business architecture needed to transform high level business processes into executable workflows

Lessons learned

- Properly encapsulate your infrastructure
- Work closely with business
- Potential identified to use infrastructure in straight-through process

SOA is everywhere in CS

- Front-to-back integration => Data quality through well-defined interfaces
- BIAN => Market-wide service standardization
- E-Commerce portal, Global Front Systems => Desktop Integration
- Financial Messaging Hubs => Standardized message formats (SWIFT, ISO 20022, FPML, …) link internal and xternal applications
- Reference data distribution => Consistent reference data across the application landscape
- Swiss platform renewal => replicated services for resilience and performance
- Datawarehouse => Managed bulk services feed data into warehouses

Where we reach the limits – Areas for research I

- Security remains a challenge
  - Secure passing of the original initiator to all services involved (in the absence of sessions)
  - Second line of defense: Application level firewalls analyzing service traffic, distributed logging of service invocation, service monitoring for unexpected use
  - Fine-grained access control
- Managing large service networks
  - Version/release management with 1000s of services and 100000s of clients, semantic compatibility
  - System Management, service-level control, fault-tolerant designs, capacity management, service replication
  - Testing of complex component networks
  - Accounting of service use
  - Development governance, balancing reuse with demand-driven development
  - Cross platform interoperability

Where we reach the limits – Areas for research II

- Semantic alignment
  - Keeping large service landscapes semantically aligned, federation
  - Semantics in dynamic service discovery
  - Systematically linking integration architecture to business architecture (information model, process model, function model)
  - Blending external standards with internal extensions
- High volume, low latency implementations
  - Market data, 100000s of messages per second, distributed to many clients, publish subscribe pattern, sub-ms latency expectation
  - Special HW, FPGA
  - Special Network-Devices
  - Simplified protocols
- Cloud services
  - Service markets
  - Domain-based interface standardization
  - Security
  - Pricing

Thank you!

QUESTIONS