



## **It's not About Technology Anymore (Including the Case for Modularity)**

**Eric Newcomer**

**Chief Architect, Investment Banking Division, Credit Suisse**

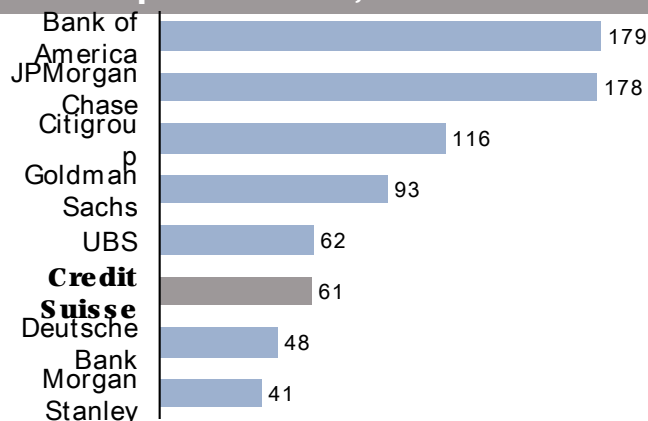
# Credit Suisse Group today – key facts

- **Global bank** headquartered in Zurich, serving clients in private banking, investment banking and asset management.
- **Registered shares** of Credit Suisse Group AG (CSGN) are listed in Switzerland (SIX) and as American Depositary Shares (CS) in New York (NYSE).
- Total number of **employees**: 49,200.
- The Group's **long-term ratings** are: Moody's Aa2, Standard & Poor's A, Fitch Ratings AA–.

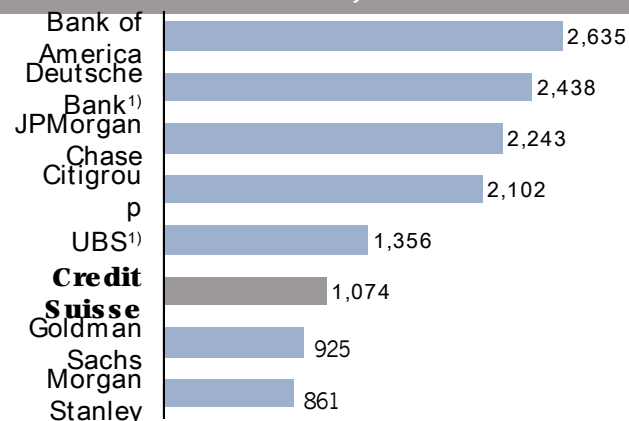


# Credit Suisse vs. peers: March 31, 2010

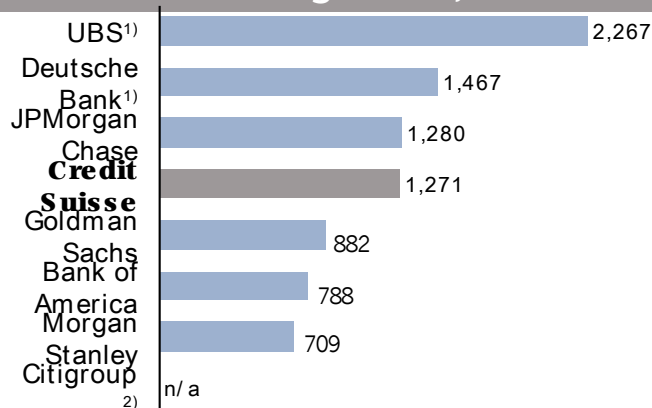
## Market Capitalization, USD bn



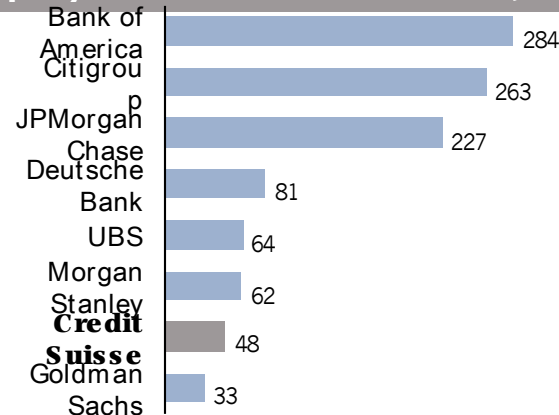
## Balance Sheet Assets, CHF bn



## Assets under Management, CHF bn




## Employees – Number of FTEs, t





1) IFRS accounting standards

2) Citigroup does no longer disclose any information on AuM

# Dynamic Business requires IT agility

- 
- Many current business trends have direct impact on the corporate IT
    - Fragmentation of the value chain (In/Outsourcing, Aggregation)
    - Mergers and acquisitions
    - Frequent strategy changes
    - Offshoring
    - Cost reduction

- 
- High pressure on IT to increase effectiveness and efficiency
    - fast reaction to new requirements in business
    - fast adaptation to varying transaction volumes
    - acceptable fixed cost

- 
- IT needs to be prepared for change
    - Create structures that allow for fast adaptation and/or extension
    - Ensure long-term cost-effectiveness

→ Establish a flexible enterprise architecture based on SOA

# IT Complexity at Credit Suisse

Today's IT systems have the following characteristics:

- Very-large-scale: in CREDIT SUISSE more than 3000 applications with more than 100,000,000 Source Lines of Code
- High complexity: large number of tightly coupled, networked components
- Aging: parts of the system are becoming obsolete and must be replaced (obsolete technology, end-of-life applications)
- High rate of change: continuous flow of new business requirements which must be implemented (Several 1000 application changes per week)
- Demanding operational quality: systems must have high reliability, good availability, sufficient security etc.

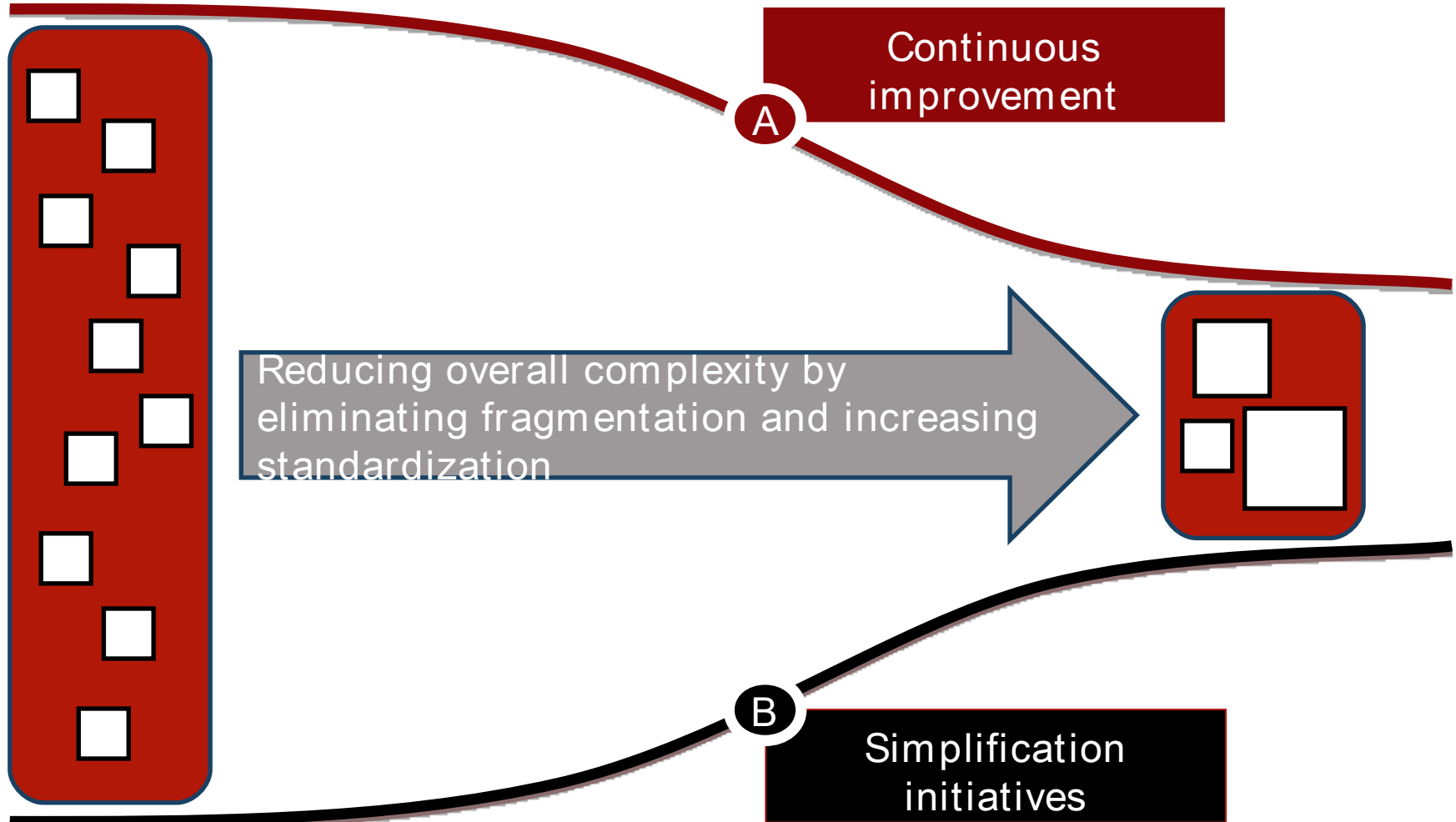
# Costs of Complexity

- Complexity increases unless something specifically is done to reduce it
- Projection is from 2010 to 2017 to produce more code total than currently exists \*
- Solution to complexity is modularity (Kirk Knoernschild et al) \*\*
- Solution to modularity is OSGi
  - Benefits of modular programming well understood for 40 years
  - Until OSGi the benefits were the responsibility of developers & architects
  - OSGi metadata and framework enforces modularity
- Modularity is a key goal of Credit Suisse's SOA efforts

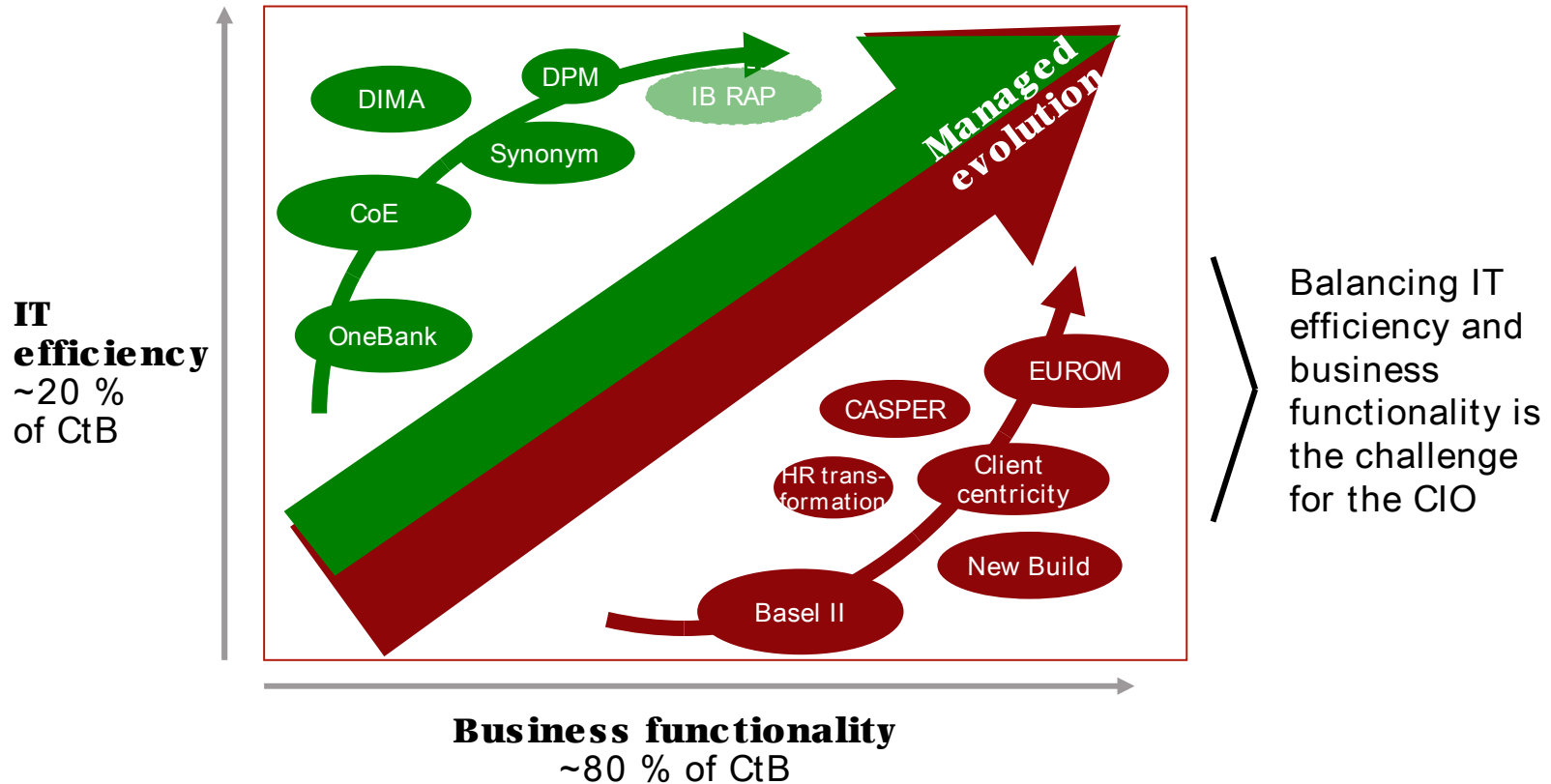
[p:// users.jyu.fi/ %7Ekoskinen/ smcosts.htm](http://users.jyu.fi/%7Ekoskinen/smcosts.htm)

[http:// techdistrict.kirkk.com/ 2010/ 02/ 17/ osgi-devcon-preview/](http://techdistrict.kirkk.com/2010/02/17/osgi-devcon-preview/)

# Our most important strategic objective for IT is to enable business growth by reducing complexity

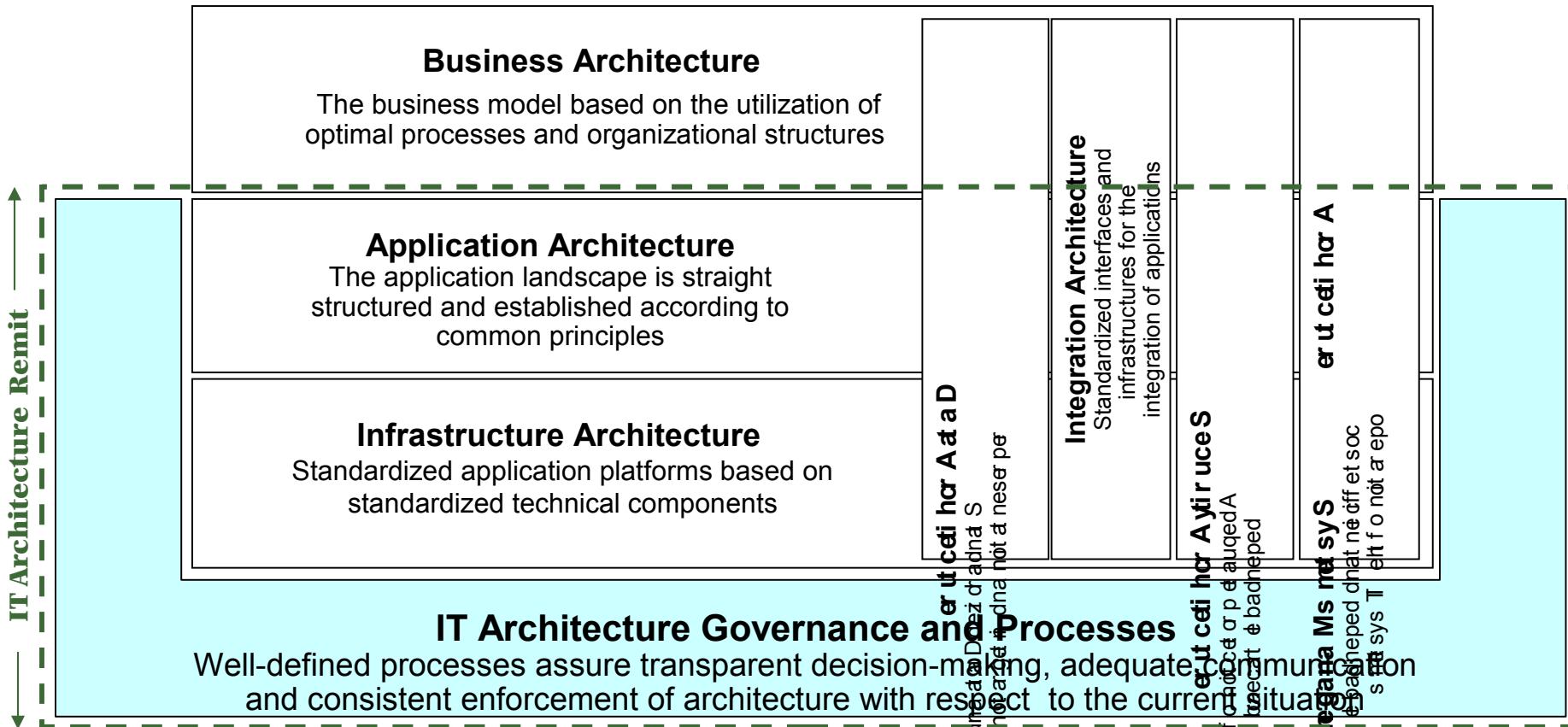


# IT Systems Strategy: Managed Evolution

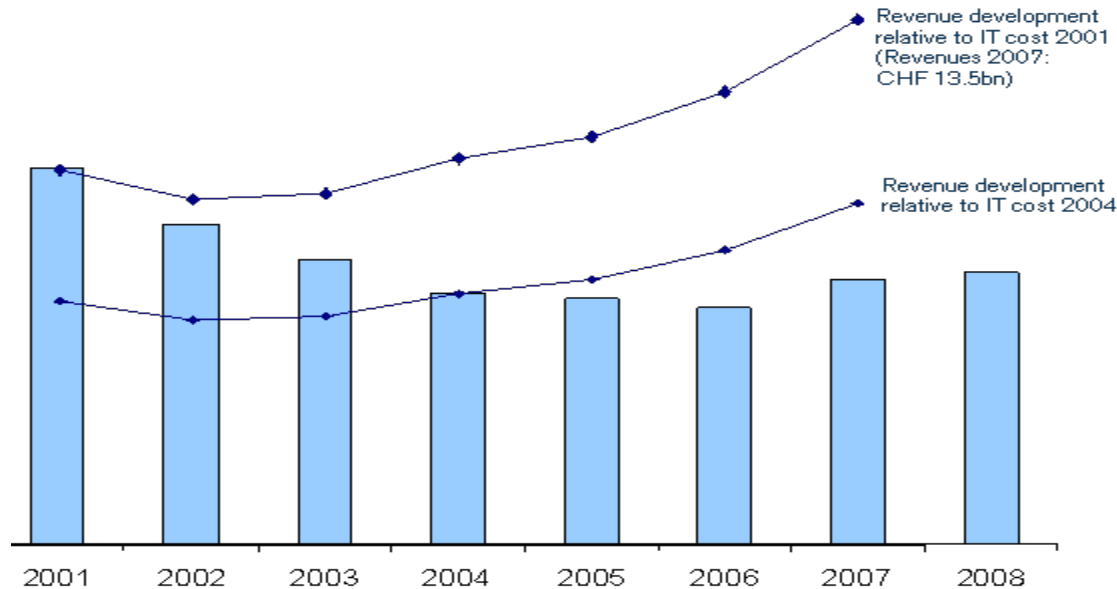


Continued investment in our infrastructure is a key cost management requirement. This process of "managed evolution" is needed to balance the two levers of investing in business functionality and IT efficiency

# Credit Suisse IT Architecture Areas



# Does IT Architecture Generate Value?



Complexity reduction  
data center

Application clean-ups (accounts, MIS,  
trading, payments, front systems, ...)

**Various technology phaseouts (OS/2, SNA, Smalltalk)**

**Application platforms (Java, DWH)**

**Service architecture, interface management**

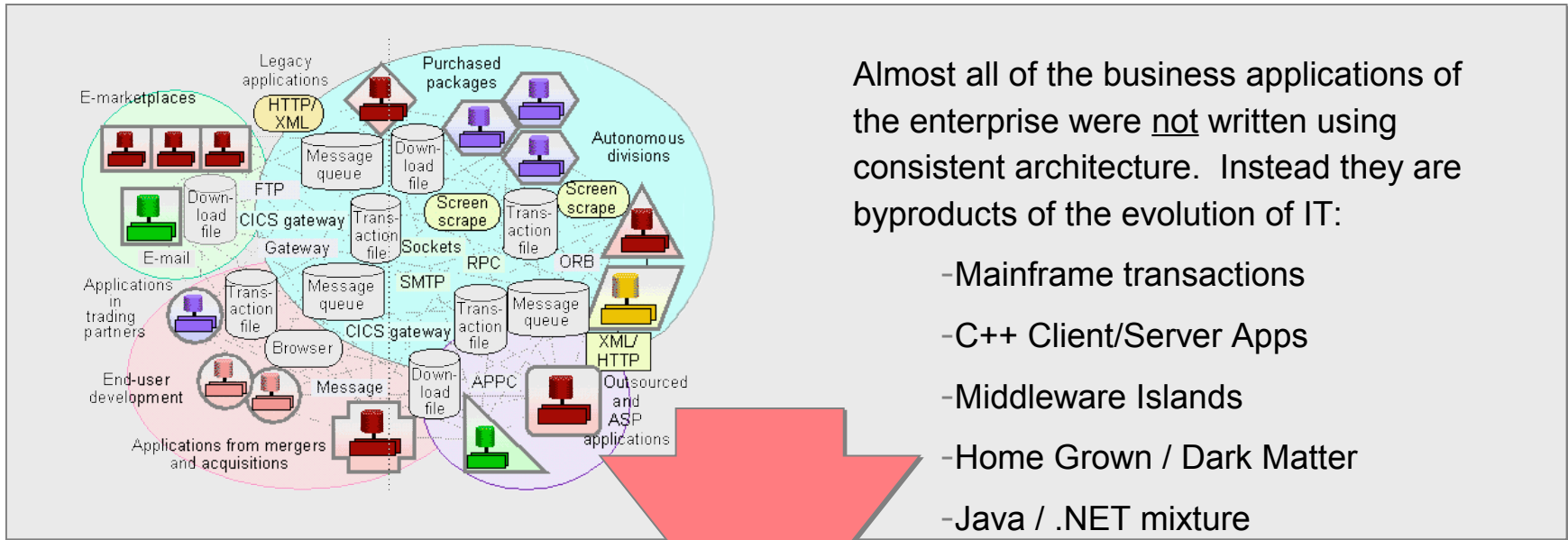
**Established architecture governance**

- Cost reduction of more than one third
- Modal split between "change" and "run" remained constant at a high level.
- Much more functionality (CRM, reporting, products, internet banking, GUIs on almost all applications, automated trading, straight-through processing, ...)
- Much more business volume
- Substantially improved stability
- Many non-architecture related initiatives (offshoring, sourcing, project portfolio management, reorganizations, ...)
- from 2007: additional cost to support growth strategy

# The emphasis shifts from functionality to efficiency

- In the early days everyone needed more features and functions to meet business automation requirements
- As the IT industry matures
  - We find we have the features and functions we need (for the most part)
  - In our products and infrastructures
- And we find, after the initial big push to automate, we have more applications than we really need
- No one was thinking about enterprise architectures
- And now we are stuck with complexity
- (You know the type of picture...)

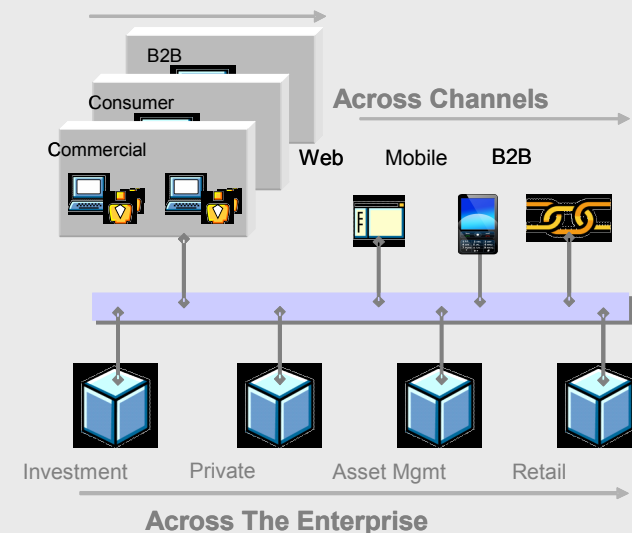




Solution: Expose and modularize existing enterprise systems as software services

- Plug-in to business applications and process flows
- Encapsulate the unique complexities
- Extend into the services foundation of today's modern software platforms

Across Business Units



# Example 1: PB – Credit Suisse Information Bus (CSIB)

## Facts

Started in 1998

More than 1000 services built up to now

All PB applications offer and/or consume services today

## Objectives

Enable Managed Evolution

Component architecture for the Swiss Platform

Reuse of core data&functionality (mainly) residing on the mainframe

## Technology

CORBA for synchronous services

WebSphere MessageBroker / MQ for messaging

Connect:Direct for files

## Footprint

About 1000 public services, 70 message publishers

280 million CORBA calls & 120 million messages delivered per month

## Example 2: PB – Global SOA

### Facts

Started in 2005

About 30 Services built up to now

Services implemented in various countries

### Objectives

Re-use the same frontend applications with different local backends

Initial driver: common global frontend application

Basis for a common front applications target architecture (FATA)

### Technology

Web Services (only synchronous communication needed)

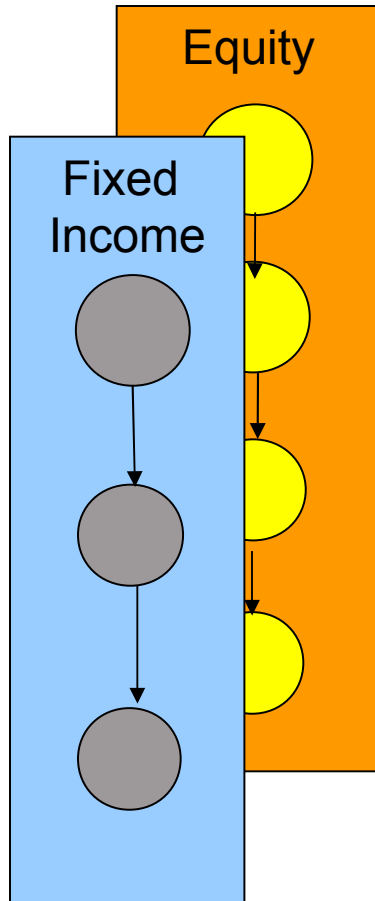
Small number of interactions today.

### Footprint

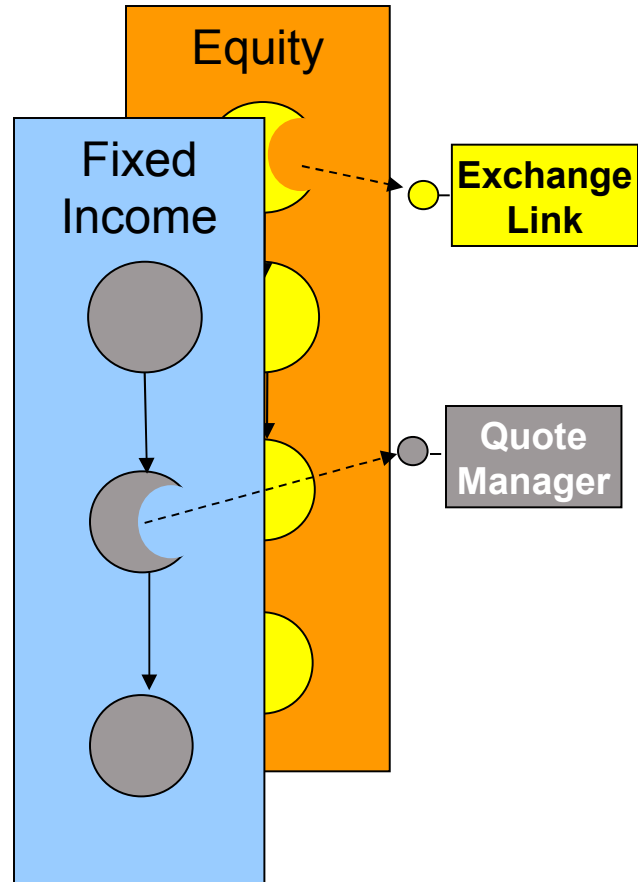
Large growth coming with important new initiatives (Global Front Components based on FATA)

# IB Approach: Moving from Silos to SOA

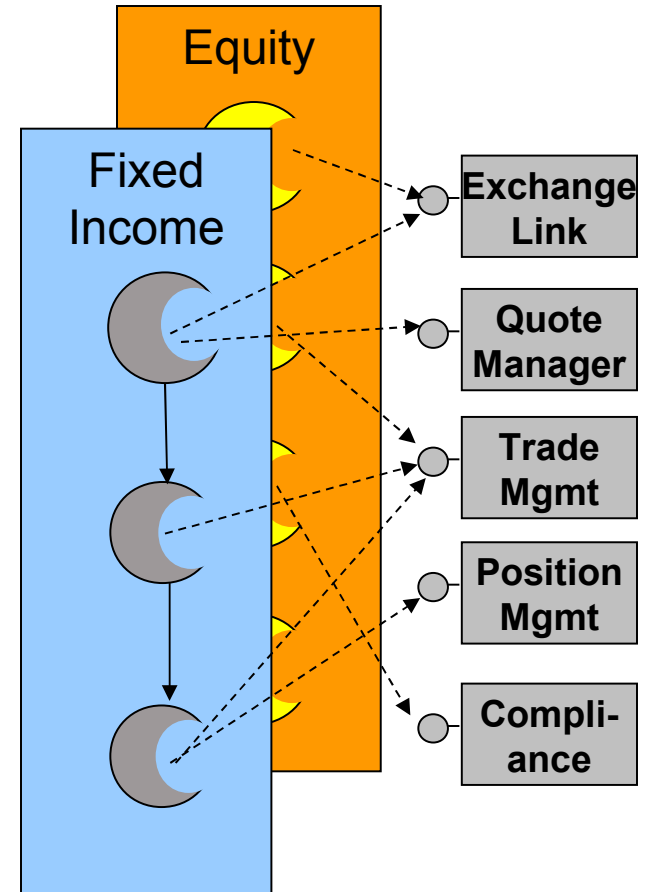
Siloed applications



Intermediate stage:  
Identify reusable components



Goal: Service-oriented architecture



# Existing Systems Need to Evolve

- They can't be replaced all at once
- Some systems have been in place for 30–40 years
- Architecture decomposes the problem
- Modules can be isolated and replaced separately
- Additional considerations include:
  - Standardizing the deployment platform(s)
  - Inventorying the applications
  - Adopting industry standards
- For example, Credit Suisse's current picture



PB Apps

CoS Apps

IB Apps

AM Apps

# More than 3000 Applications... in the Common Domain Model

## 5: Communications & Collaboration

Communication and Access (CHA) 14727 0 0

Street Side Interfaces (SSI) 7740 1 1

Business Partner Applications (BPA) 19 1 0 1

Enterprise Content Management (ECM) 8634 1 2

Financial Market Information (FIN) 65152 0 0

11772 6 49

Rolling and Reporting

Regulatory, Risk and Liquidity (RRL)

18 5 0 0

Accounting, Reporting

Accounting Core (AOC)

60 7 1 10

Financial Accounting (FAC)

2: Finance, Investment & Sales

Wealth Management & Advisory 92 7 5 1

Credits and Syndication 85 58 0 1

3: Trading and Markets

Trading (TRA) 146567 6 4

Order and Trade Management (OTM)

Product Control 2613661 6

4: Cash and Asset Operations

Payments 6335 0 0

Single Accounts 3821 1 0

Settlement and Clearing 69189 0 1

Custody 2822 0 0

Corporate Actions 2117 0 1

1: Partners

Customer & Partner (CUS) 6865 0 7

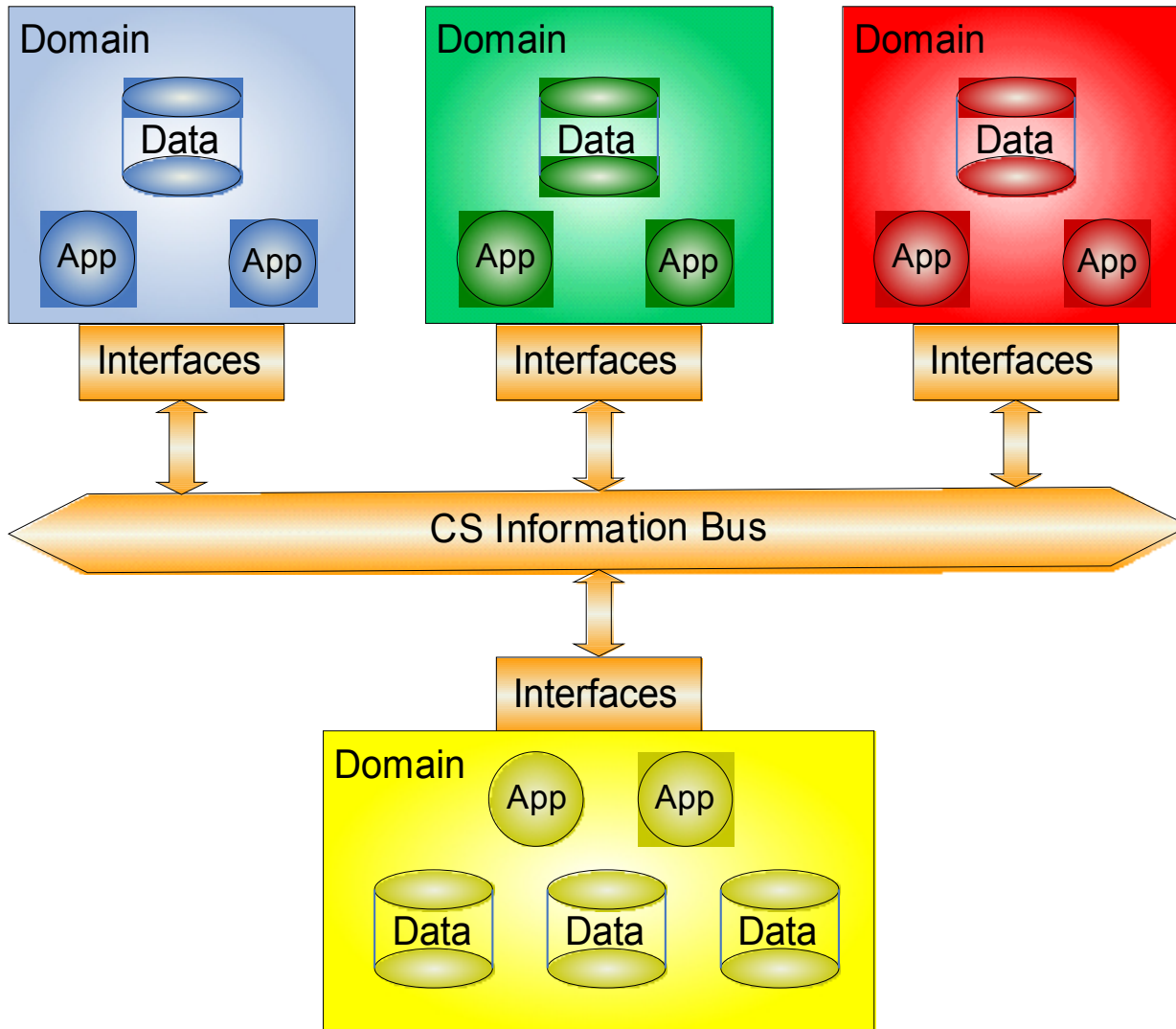
Customer Relationship Management (CRM)

7: Enterprise Common Services

2291 1 38

80105 2 2

# Domains are high-level components coupled using services

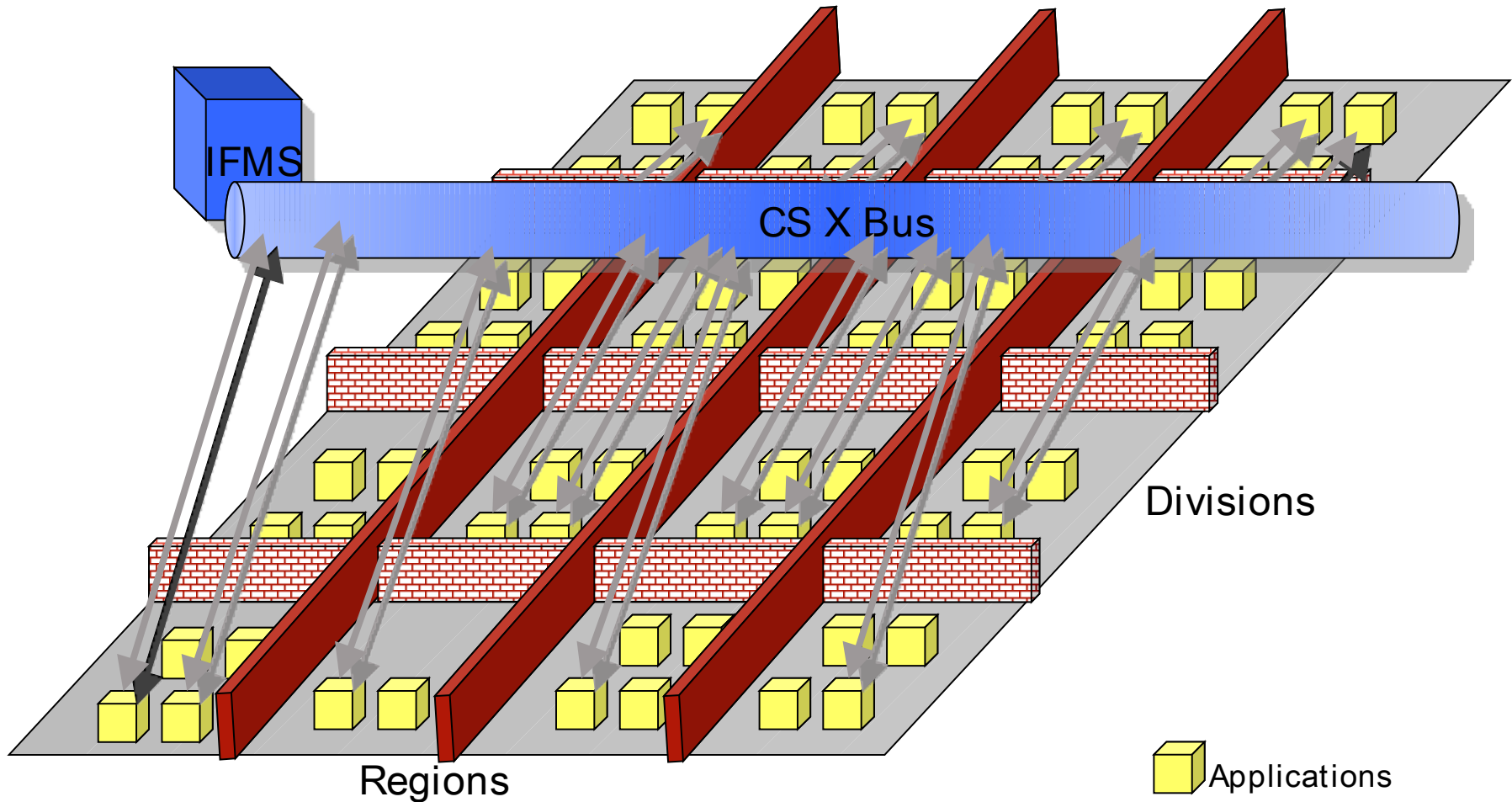


# Supporting Tool: Interface Management System (IFMS)

- Component owners need to **design** the interfaces of their components (planning and design)
- Developers need a **generator** producing source code for using the interfaces
- Service users need a **catalogue** with powerful search functions so they find the right functionality



# Target state: Credit Suisse eXchange Bus (CSXB) for federated SOA



# Summary

