



Flexible Insurance Systems

Part 2: Flexibility

What is Flexibility and how can you provide it?

Agenda



- Flexibility defined
- Kinds of Flexibility in an Insurance System
 - Product Flexibility
 - Business Process Flexibility
- Other Subsystems that need Configuration
 - Authentication
 - Printing Subsystem
- Wrap Up: Alternative Ways to provide Flexibility

Flexibility defined



We call a system flexible if it can be adapted to changing needs quickly and at moderate costs

- Problem: Which needs are going to change – you are never going to be able to anticipate whatever kind of change is possible. You have to know them in advance => contradiction
- If you do not balance flexibility and performance versus cost, you will easily end up with an inflexible or hyper flexible system.

Flexibility comes with a bunch of trade-offs

Flexibility often stands against

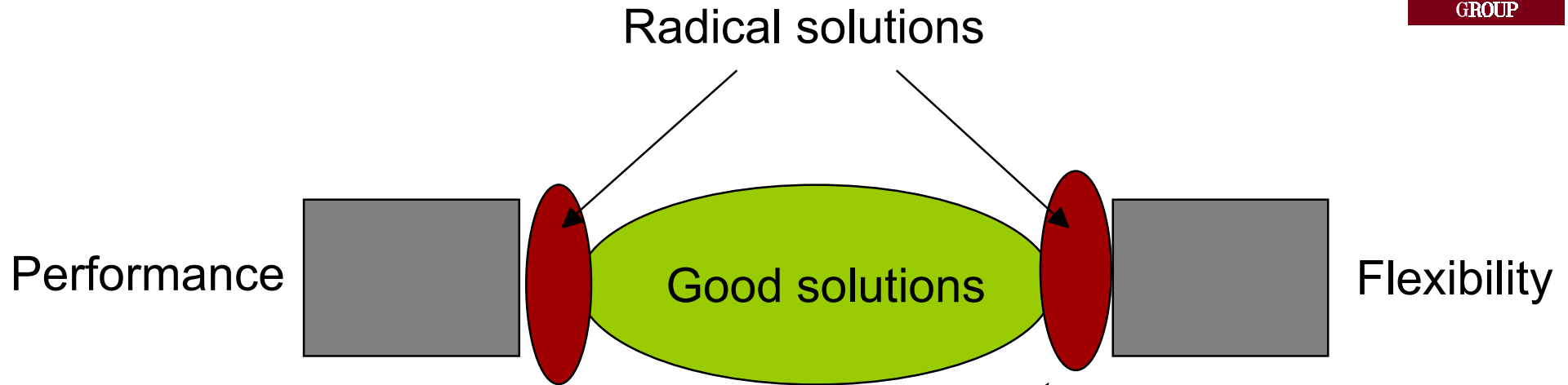


- Performance
- System construction costs
- Modularization and secrecy of objects
- Cheap and safe configuration management
- Understandability, Simplicity
- Ease of construction



See
the Pitfalls

Flexibility comes with a bunch of trade-offs

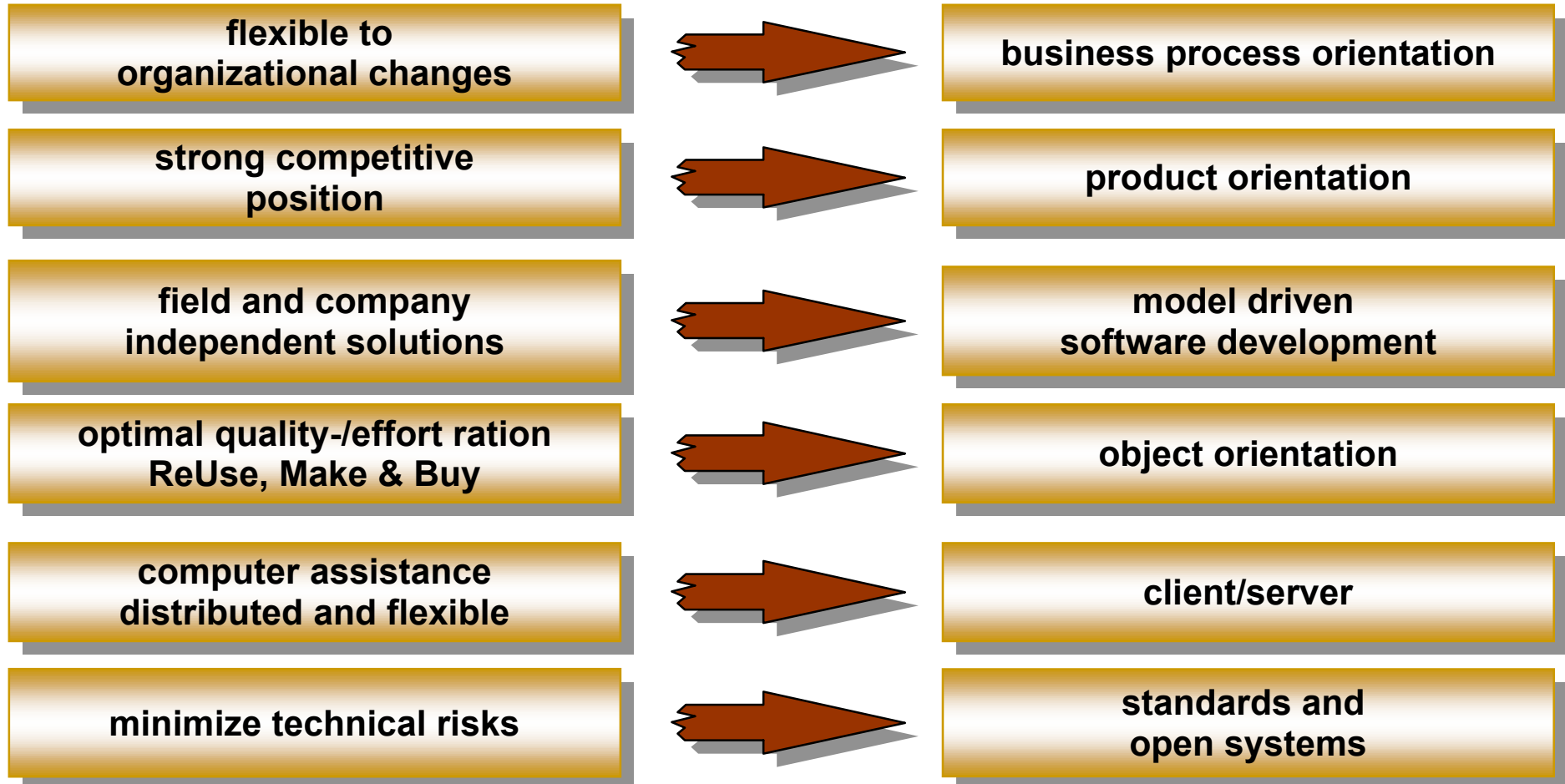


Beware of the Hyper Flexible System

See
the Pitfalls

Example

Phoenix Architectural Principles



Most Important Kinds of Flexibility in an Insurance System



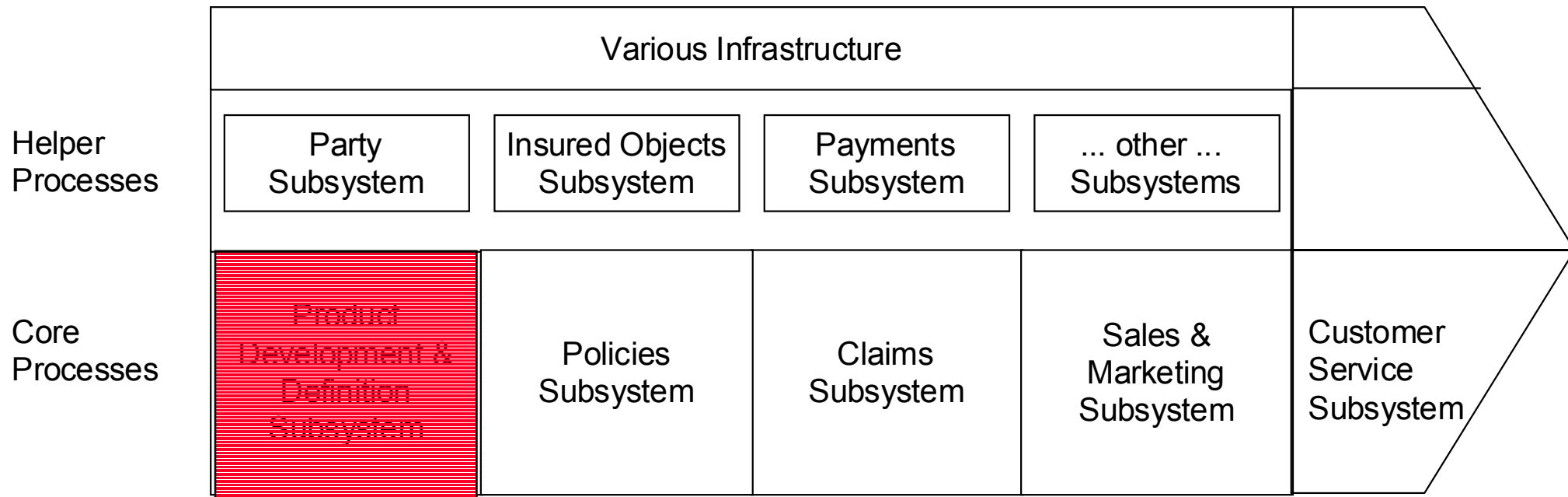
- Product Flexibility
- Business Process Flexibility

Product Flexibility



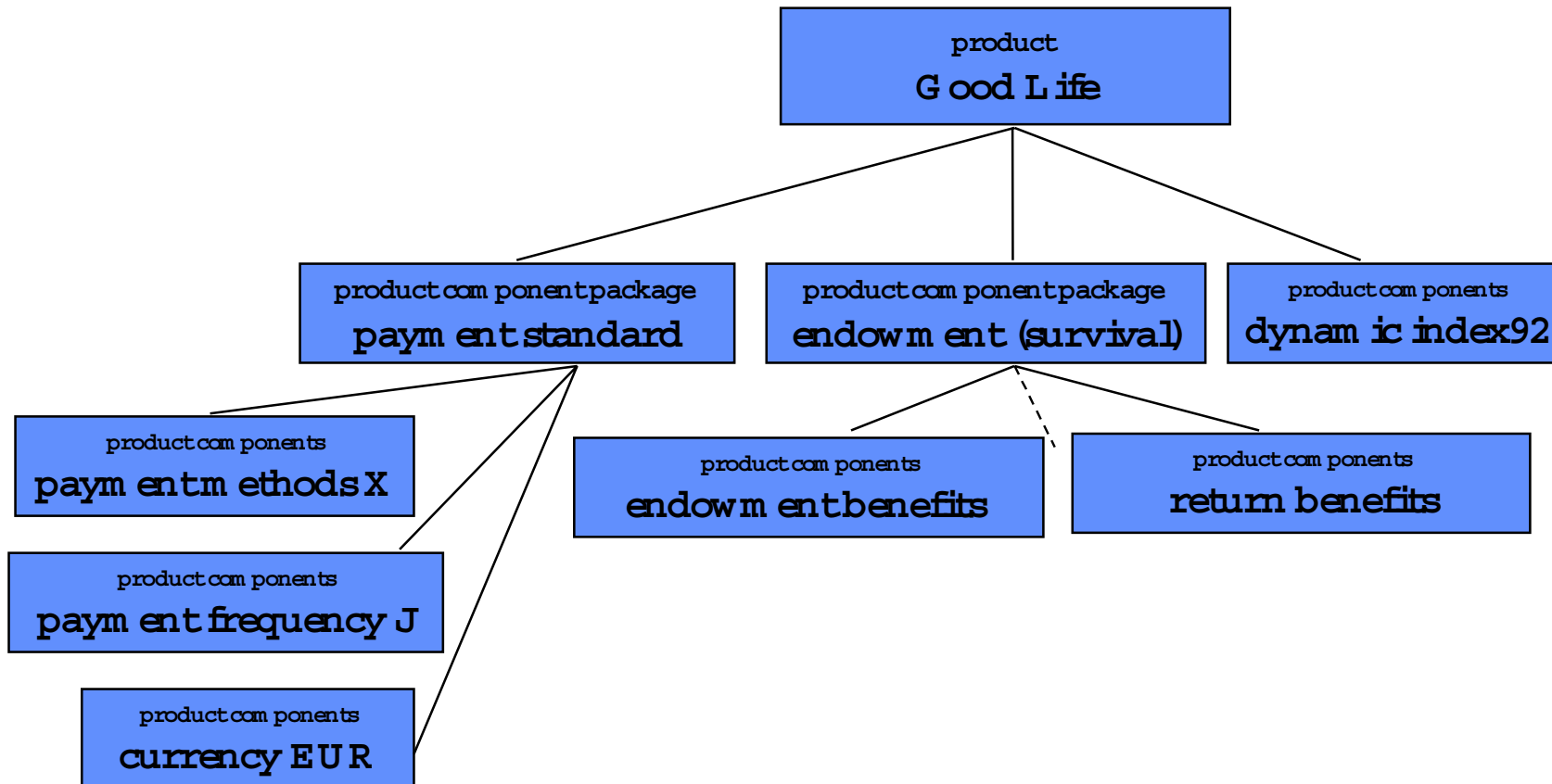
- Basic Idea - The Product Server Pattern
- Outlook towards Product Modelling - OEI Pattern

Product System



Product System

Example Product



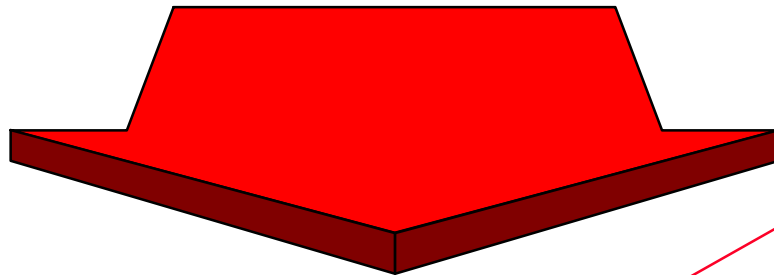
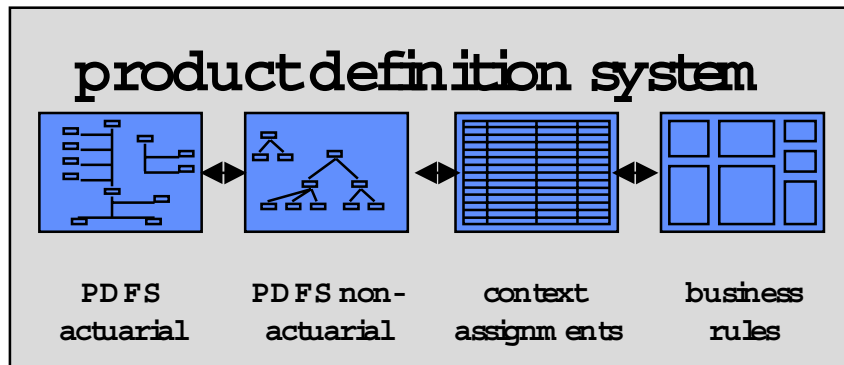
Architectural Principles

Product Orientation

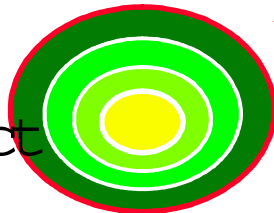


- Phoenix
 - transfers structural information about products from software into definitions
 - is flexible, because business departments can change definitions using administration tools (as long as no fundamental changes of products occur).
 - is stable, because operative systems interpret definitions
 - is individually configurable and parameterized by each subsidiary

Product System Product Definition



Product

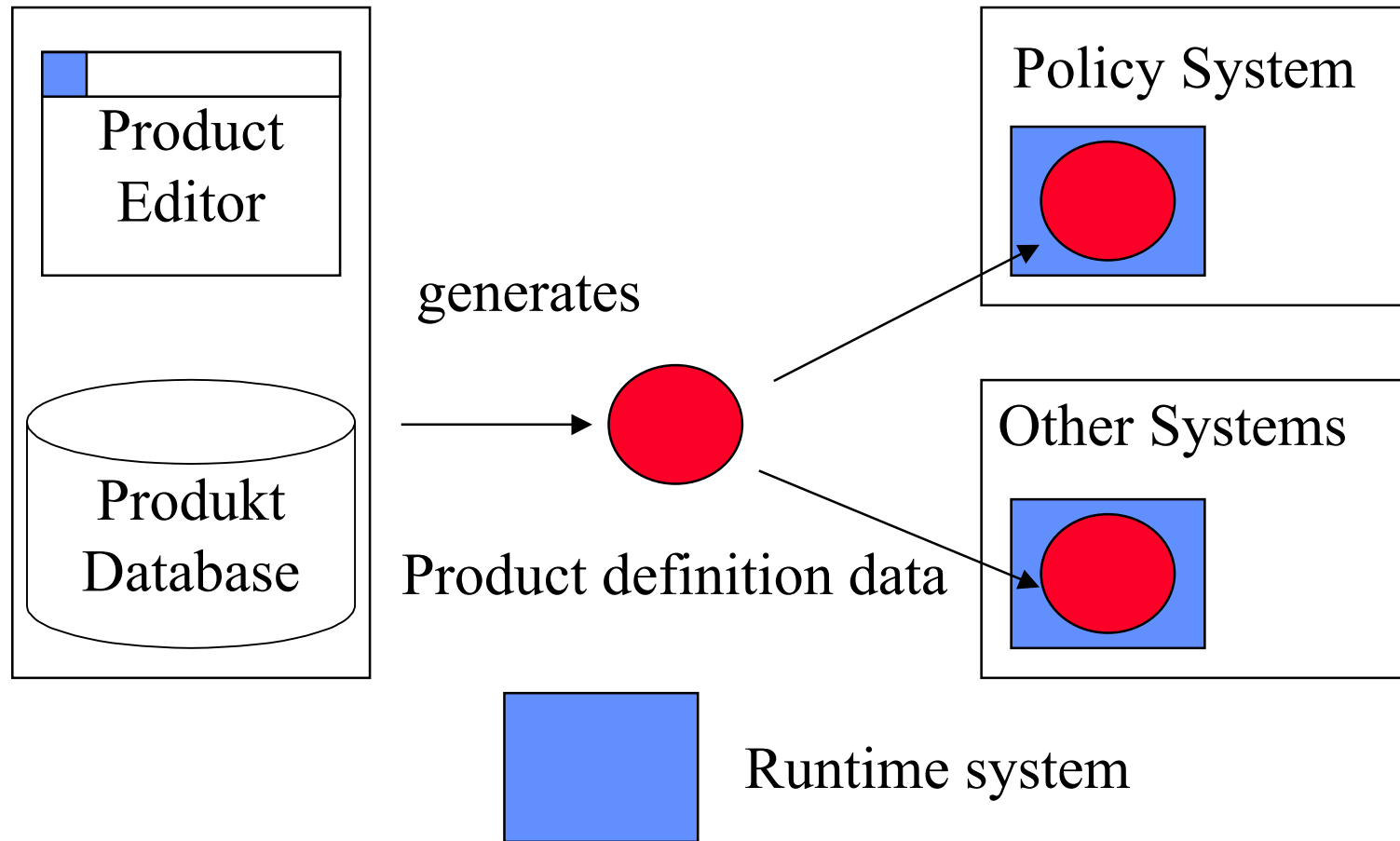


Applications



...

Structure of the Product Server Pattern





Business Process Flexibility

Business Process Flexibility



- The Goals of Business Process Modelling and Business Process Reengineering
- Technical Ways
 - Using no Technology
 - Using Email & Similar Systems
 - Using Production Workflow Systems

Goals

Business Process Reengineering



- Create business processes that provide maximum service for your customer at minimal cost
- Or other goals like
 - minimum customer response time
 - minimal costs
 - ...

Ways Business Process Reengineering



- If appropriate defy Taylorism
- Use Pareto principle (80/20 Rule)
- Streamline mass cases
- Reintegrate process steps for mass cases
- Assign special cases to special people

Business Process Modelling

Why



- Provide models and documentaion of your newly designed business processes
- Using Tools like
 - Adonis (BOC)
 - ARIS (Scheer)
 - ... others

Common Misconception



- Business Process Reengineering does not imply using a Business Process Modelling tool
 - you may as well use paper and a pencil if you redesign just one process
- Thinking in Business Processes does not imply using a Production Workflow Manager
 - This is a very common and costly error
 - Production Workflow is only suitable for a very narrow problem domain - if you don't have the problem, applying the solution will be pretty senseless and expensive

Using Production Workflow Systems



- Business Process Modelling
- Business Process Execution Engine
- Link 1: Importing Business Process Definitions
- Link 2: Calling Activities

Other Subsystems that can use Configuration



- Authentication
- Printing

Technical Infrastructure Authorization



Berechtigungsadministration (Knoten: Generaldirektion)

Administration Komponente Tools Ansicht Hilfe

ID: AX00076 Positionieren

- + Hartmann Hans 2.Account
- + Handler Rudolf 2.Account
- + Simlinger Maria 2.Account
- + Eder Josef 2.Account
- + Pils Hannes 2.Account
- + Wimmer Erich 2.Account
- Blüml Werner 2.Account
 - Chefsachbearbeiter
 - POA Chef
 - Sachbearbeiter
 - PC Print Client
 - Huber Hans
 - Chefsachbearbeiter

Rollen

Symbol	Typ	Id	Name	Gültig v
	Berechtigungsroll	BREADM	Berechtigungsadministrator	17.12
	Berechtigungsroll	CHEFSB	Chefsachbearbeiter	17.12
	Berechtigungsroll	DRMINFO	DRM Beauskunfter	17.12
	Berechtigungsroll	DRMENTW	DRM Entwickler	17.12
	Berechtigungsroll	DRMENTVA	DRM Entwicklungsverantwortlicher	17.12
	Berechtigungsroll	DRMFRGVA	DRM Freigabeverantwortlicher	17.12
	Berechtigungsroll	FSB	Filialsachbearbeiter	17.12
	Berechtigungsroll	INKASSSB	Inkassosachbearbeiter	17.12
	Berechtigungsroll	PCROLE	PC Print Client	19.12
	Berechtigungsroll	POACHEF	POA Chef	17.12
	Berechtigungsroll	POALEV1	POA Level 1	17.12
	Berechtigungsroll	PRSEADM1	PRS Administrator	17.12
	Berechtigungsroll	SB	Sachbearbeiter	17.12
	Berechtigungsroll	SPAKZEPT	Spezialist Akzept	17.12

Speichern

Bereit | SYSBRE1 | Ansicht: BENUTZER

Wrap Up:

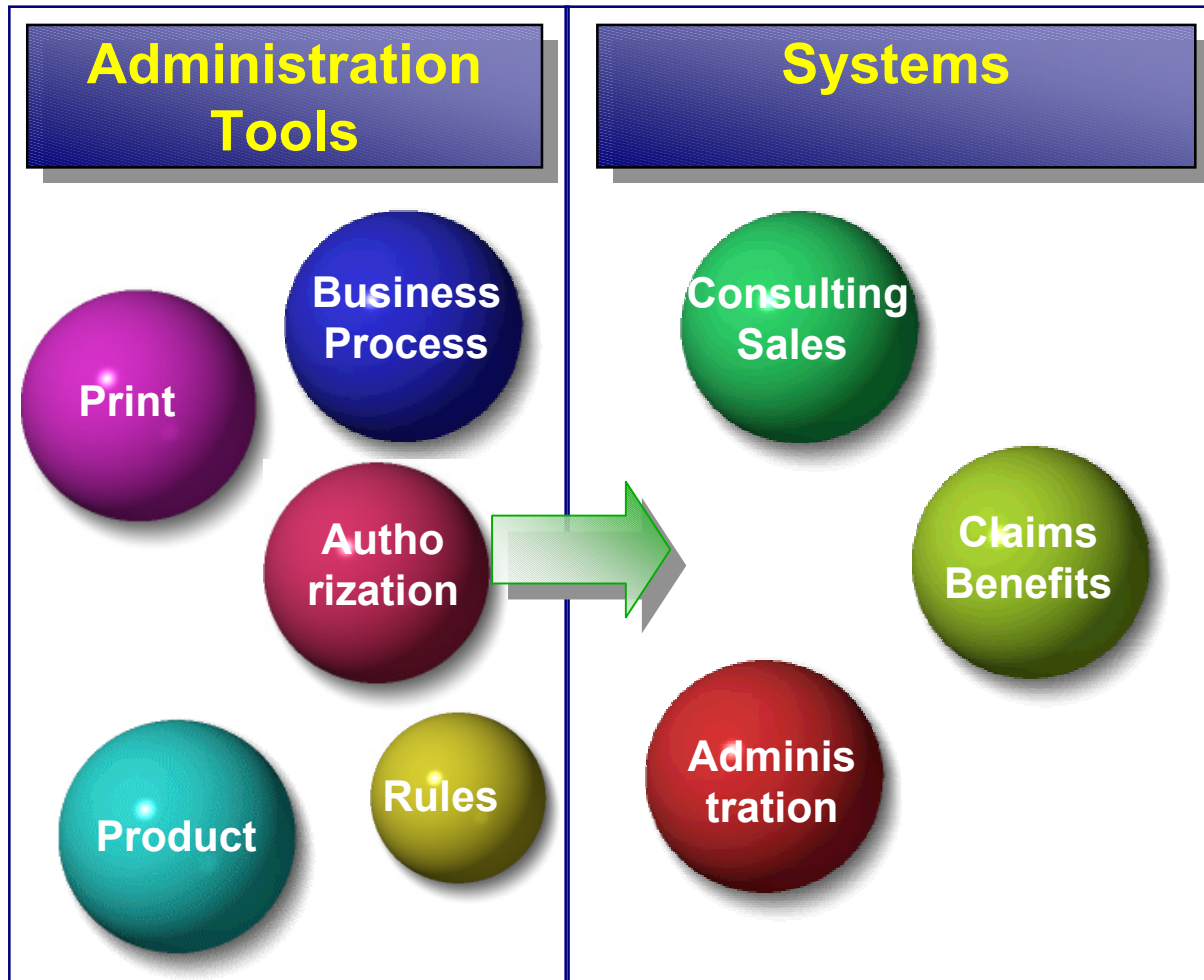
Alternative Ways to provide Flexibility



- **Rapid prototyping** and fast to use programming environments
Just hardcode it, e.g. in Smalltalk
- Speed up things by using **frameworks**, promoting reuse and having a **good flexible architecture**
- **Meta-system approaches**, creating domain specific definitions systems (languages)
- Any other form of **Interpreters** (Workflow, Authentication System)
- **Open Implementation** (sorry – no practical experience with that)

Flexibility via Meta-Systems and Interpreters

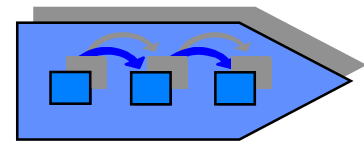
Model Driven Development



Customizing of Phoenix Life can partly be done by definition using administration tools

Goal: Higher level abstraction than programming

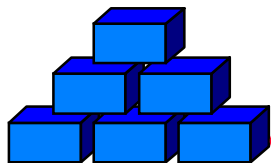
Example: Phoenix Technical Infrastructure Integration of process and product influences



Business process

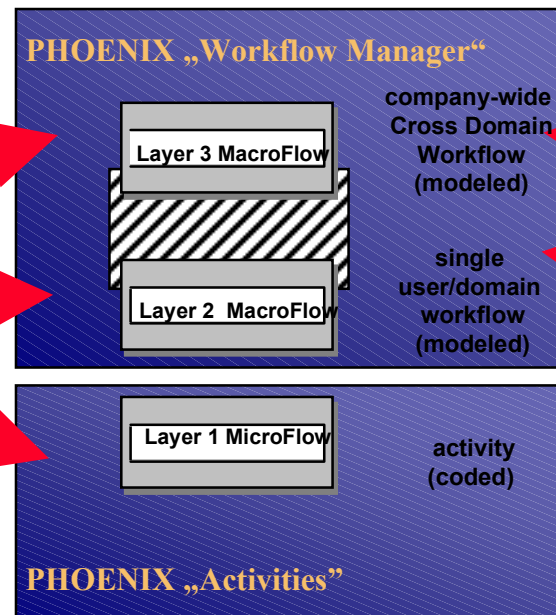
☞ push

- ☞ process variables
- ☞ visibility, pre- & post conditions
- ☞ limited branching



Insurance product

- ☞ field configuration
- ☞ plausibility tests
- ☞ options



☞ forwarding

☞ static authorization



Authorization

☞ dynamic authorization