

Architecting and Standardization

by *Gerrit Muller* Embedded Systems Institute
e-mail: `gerrit.muller@embeddedsystems.nl`
`www.gaudisite.nl`

Abstract

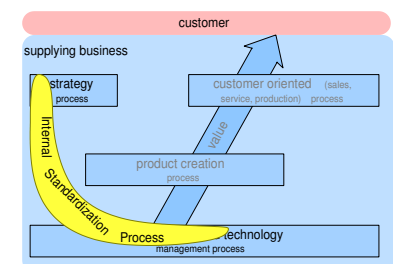
Many products today are developed for highly dynamic markets while the products and functions get more and more integrated. The product and service realization is based on fast changing technologies that come together in complex value chains. The challenge for modern companies in innovative domains is to survive in this dynamic world.

In this paper we explore the contribution of architecting and standardization to the company success. We look at the *why*, *when*, *who* and *how* questions of standardization and at the role of architecting in the standardization process.

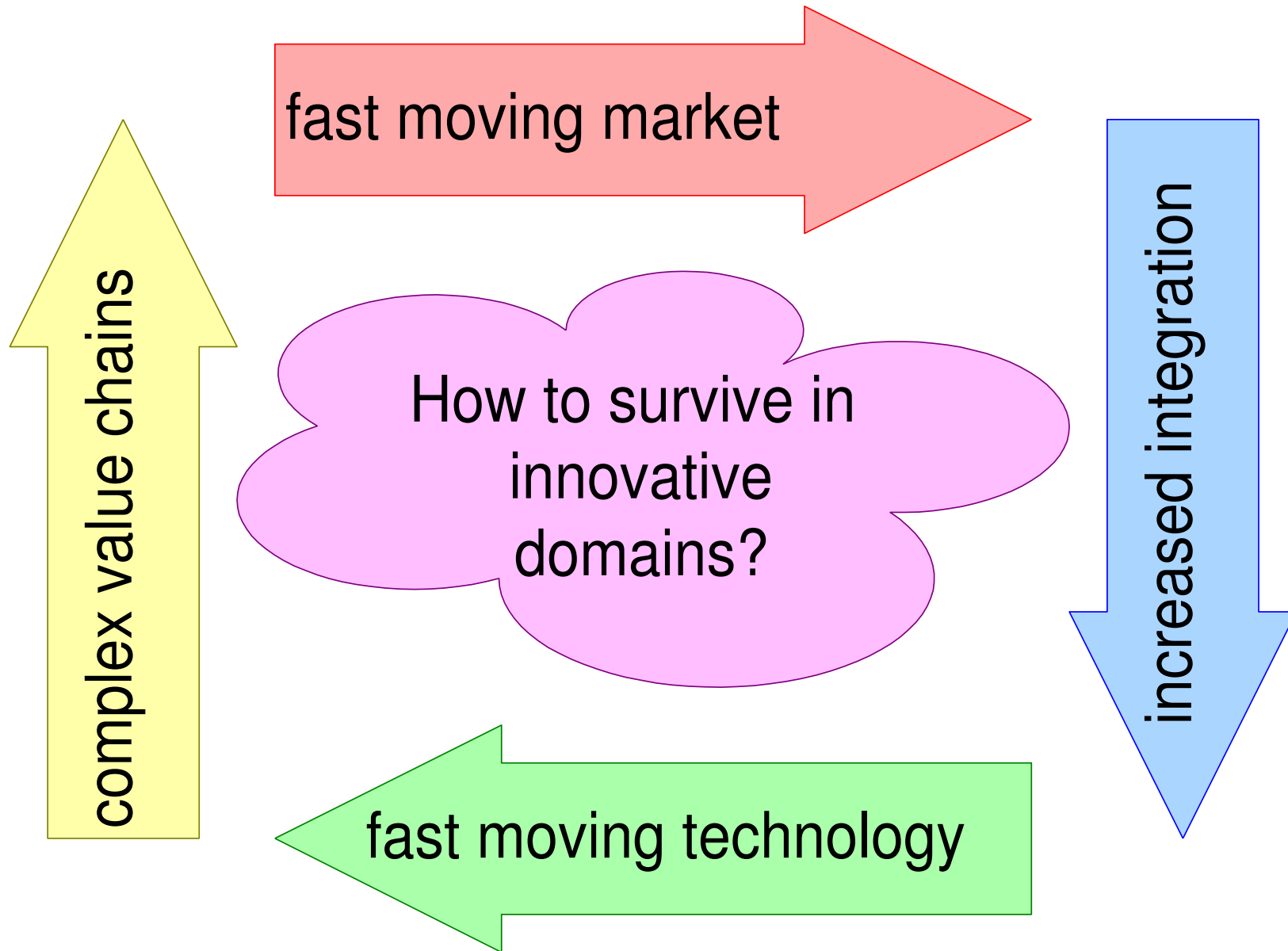
Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

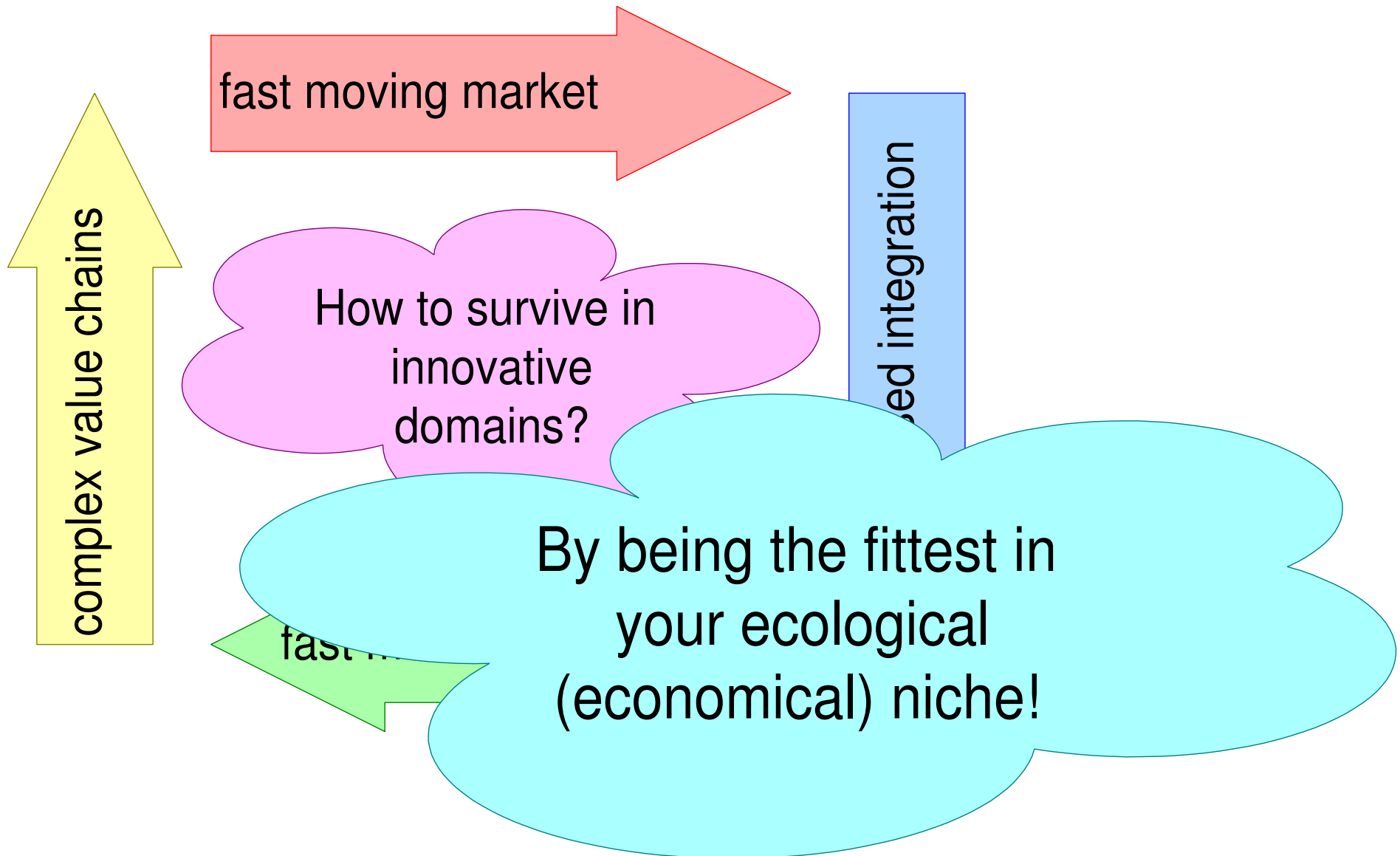
July 1, 2011
status: draft
version: 1.0



Problem Statement



That is easy...



Postulated Solution

1. employ skilled system architects
2. apply an agile system architecting process
3. determine the right subjects and moments for standardization
4. apply a sensible standardization process

Figure Of Contents™

How to survive in
innovative domains?

standardization

what

why

how

when

who

How to survive in
innovative domains?

standardization

what

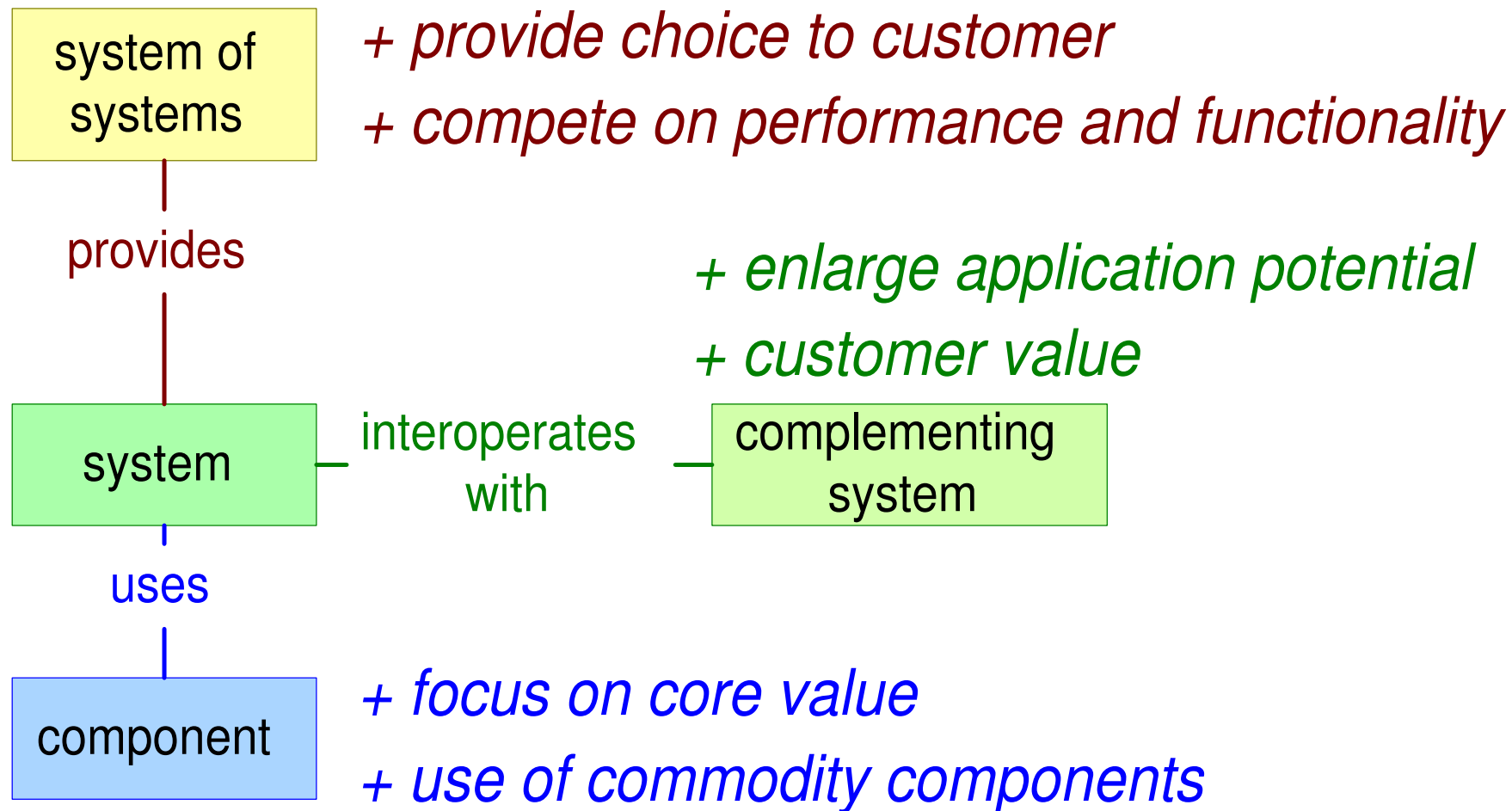
why

how

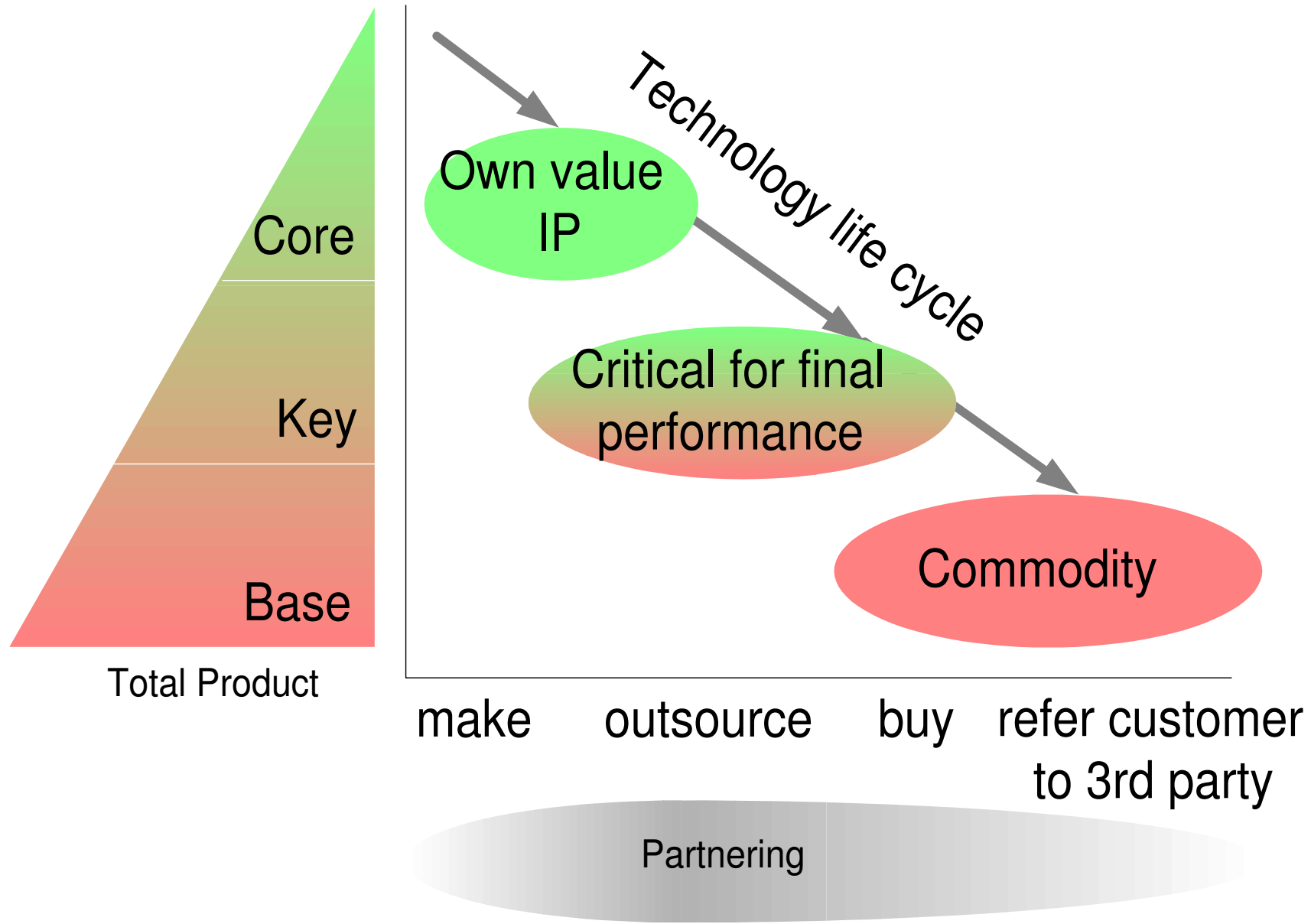
when

who

Classification of Standardization Tactics



Focus on Core; not on Key or Base Technology?



How to survive in
innovative domains?

standardization

what

why

how

when

who

When to Standardize

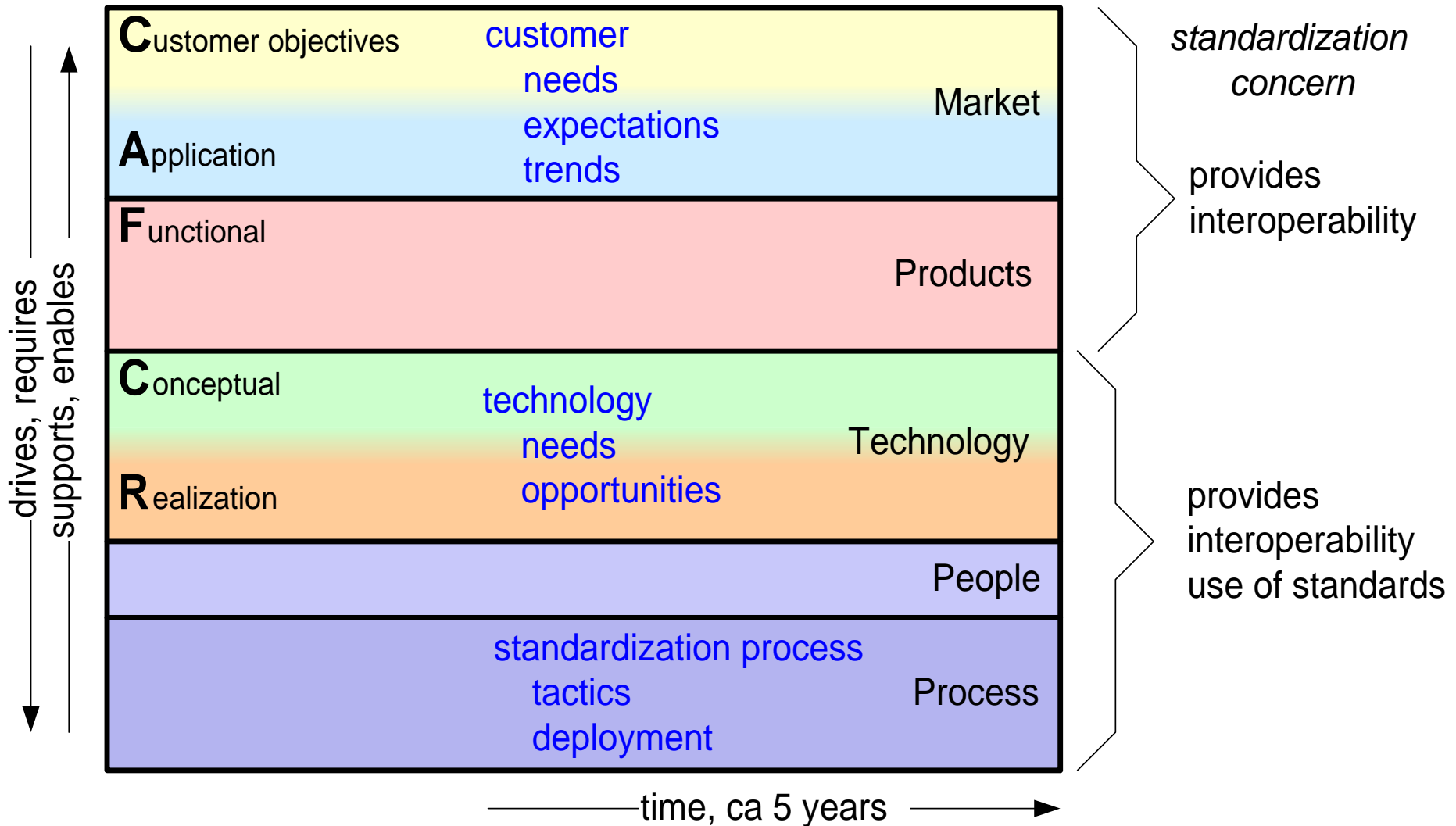
too early ← right moment → too late

problem is understood
domain structure is clear
broadening set of stakeholders
technology is ripe

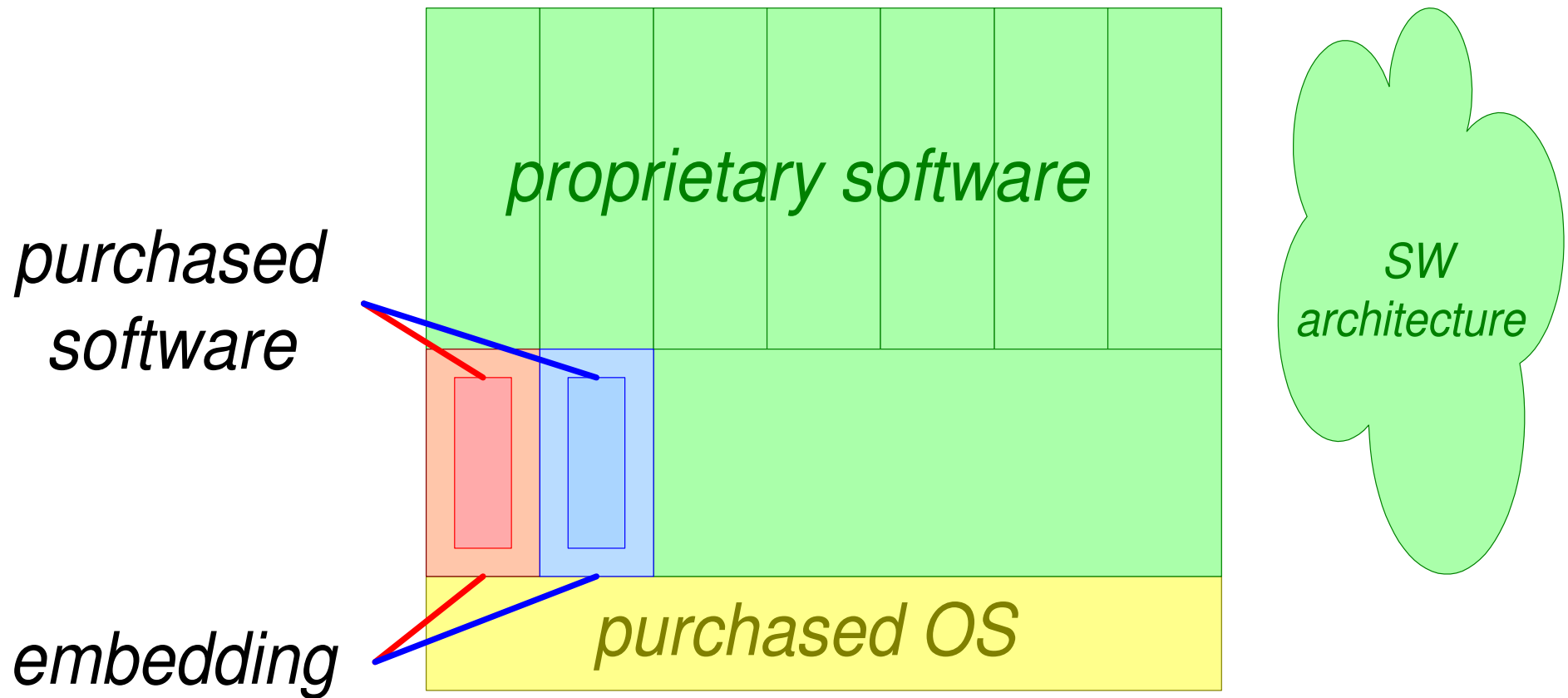
requirements unknown
technological compromises
loss of competitive edge
insufficient and uncertain facts
wrong expectations
intuition not calibrated

caught in proprietary legacy
poor interoperability
customer demands standards
focus on key i.s.o. core
market does not take off
(Metcalfe's law)

Roadmapping as Tool



Purchased SW Requires Embedding



Embedding Costs of Purchased SW

- Installation
- Configuration
- Customization
- Start up, shutdown
- Specifications
- Interface to application SW
- Exception handling
- Resource allocation and monitoring provision
- Resource tuning, see above
- Safety design
- Security design

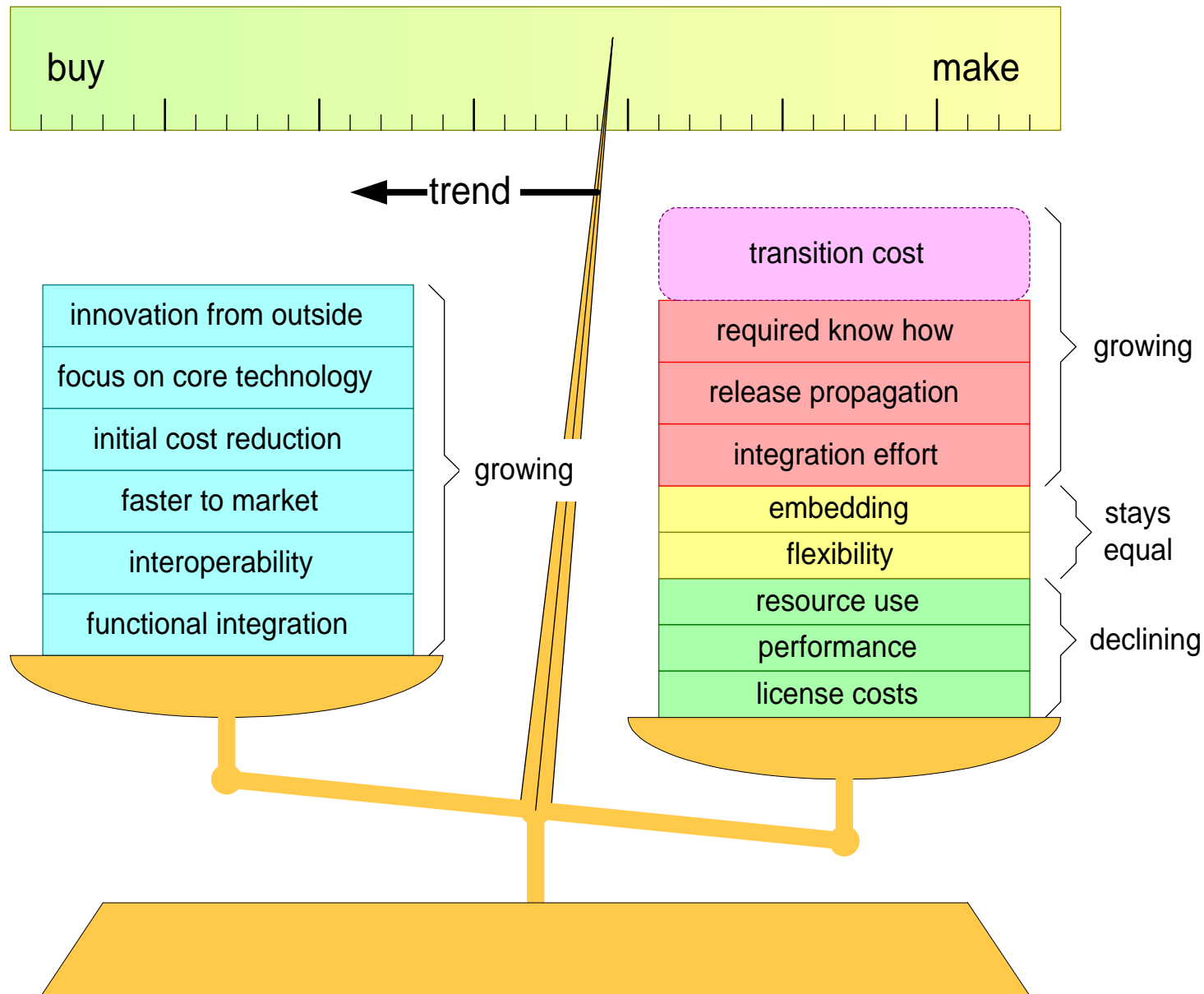
functional
system design
sw design

add semantics level
use of appropriate low level mechanisms
match to high level mechanisms:
- notification, scheduling
- job requests, subscriptions

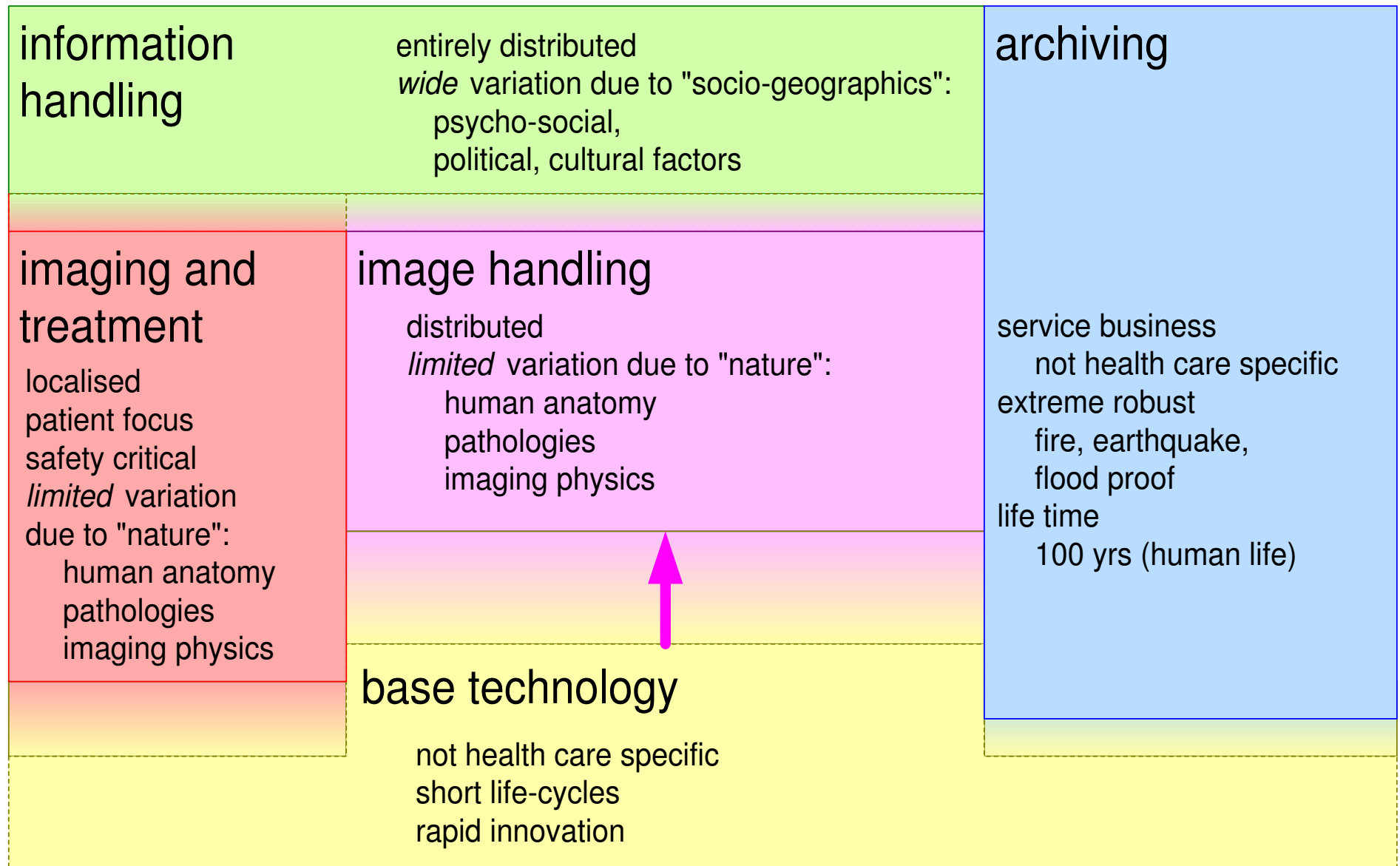
System monitor
Error propagation
Logging

CPU
Memory
Disk

Balance of Considerations and Trends



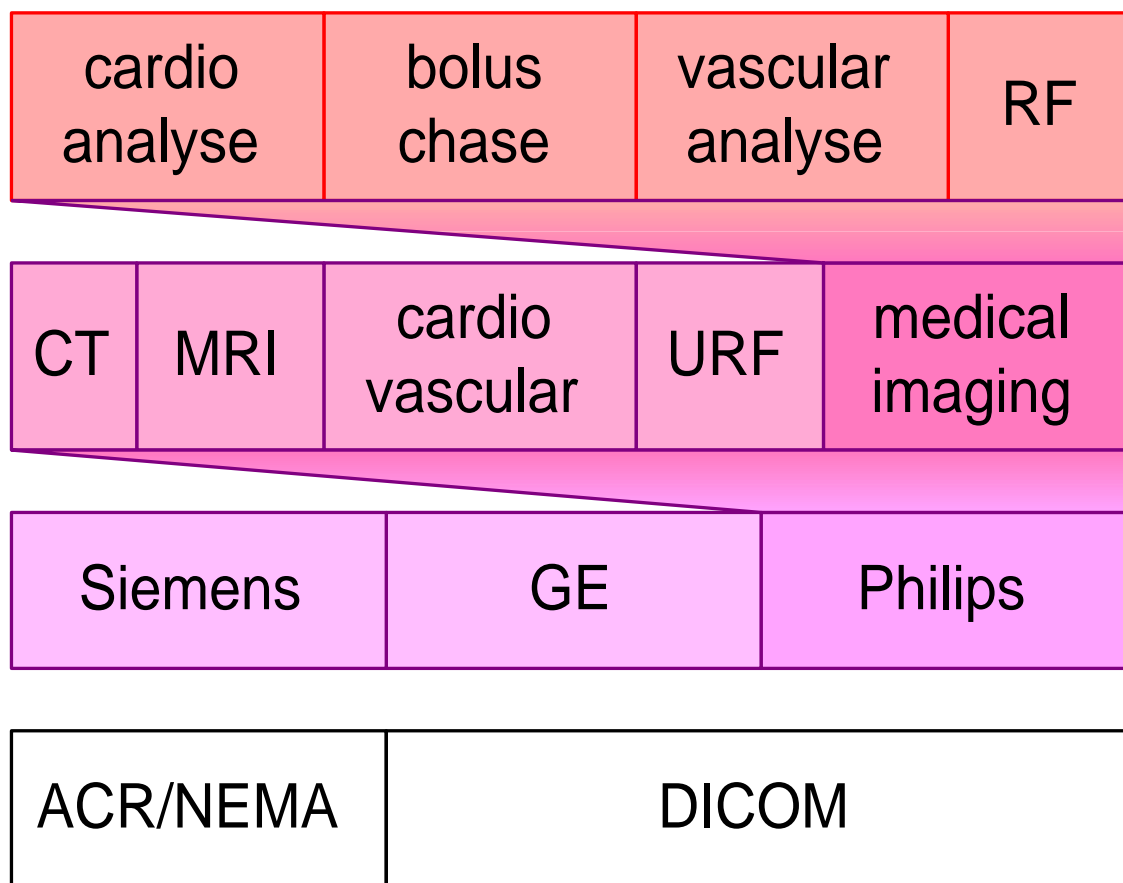
Example of Lifecycle Reference Model



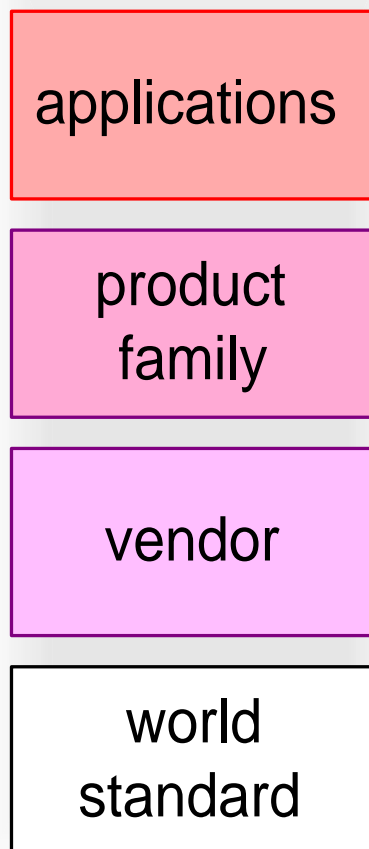
Evolution from Proprietary to Standard

high innovation rate

global standardization
takes more than 5 years



legend



high interoperability

How to survive in
innovative domains?

standardization

what

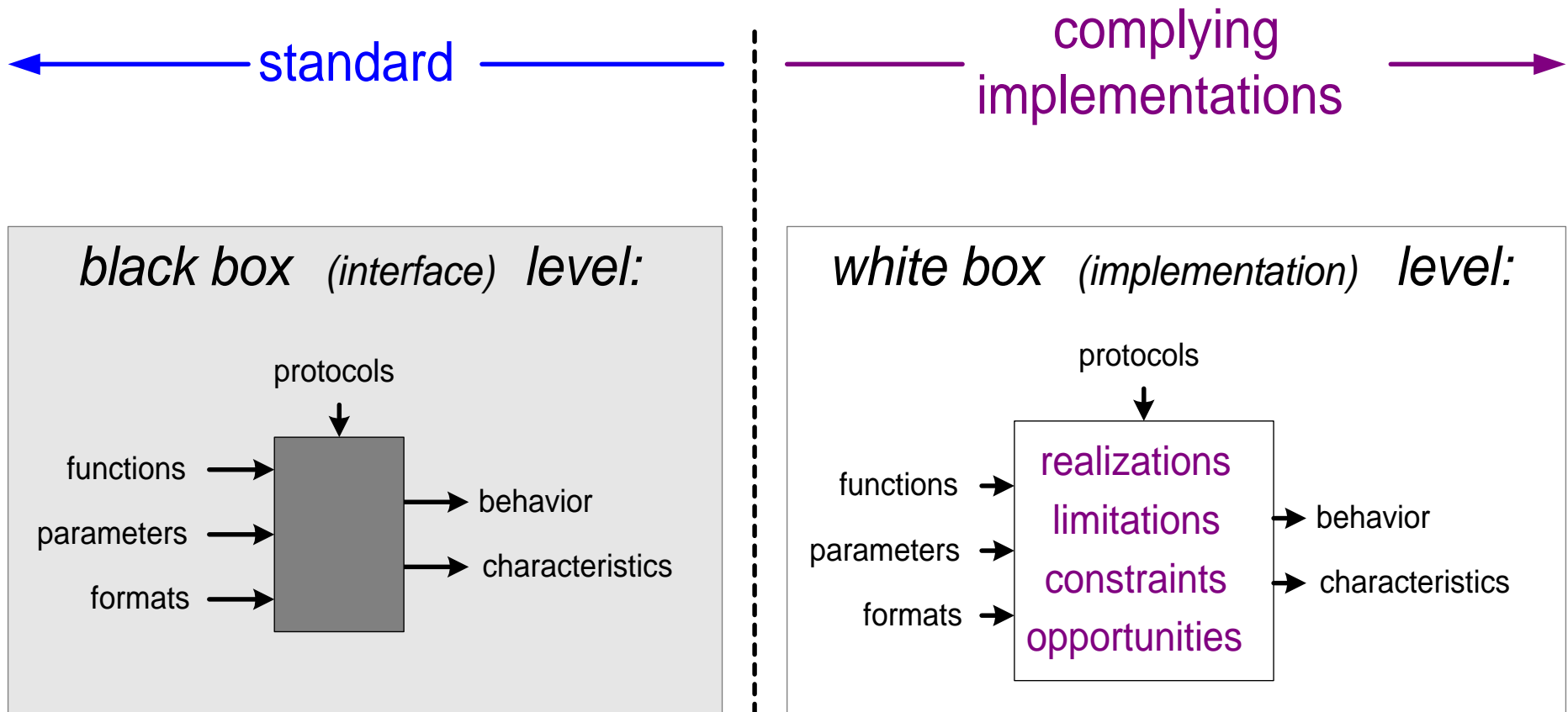
why

how

when

who

Standards describe **what**



white box know how:

current and future realization:

design choices

technology capabilities

domain concepts

limitations

constraints

opportunities

what needs to be defined

functions

parameters

formats

protocols

behavior

characteristics

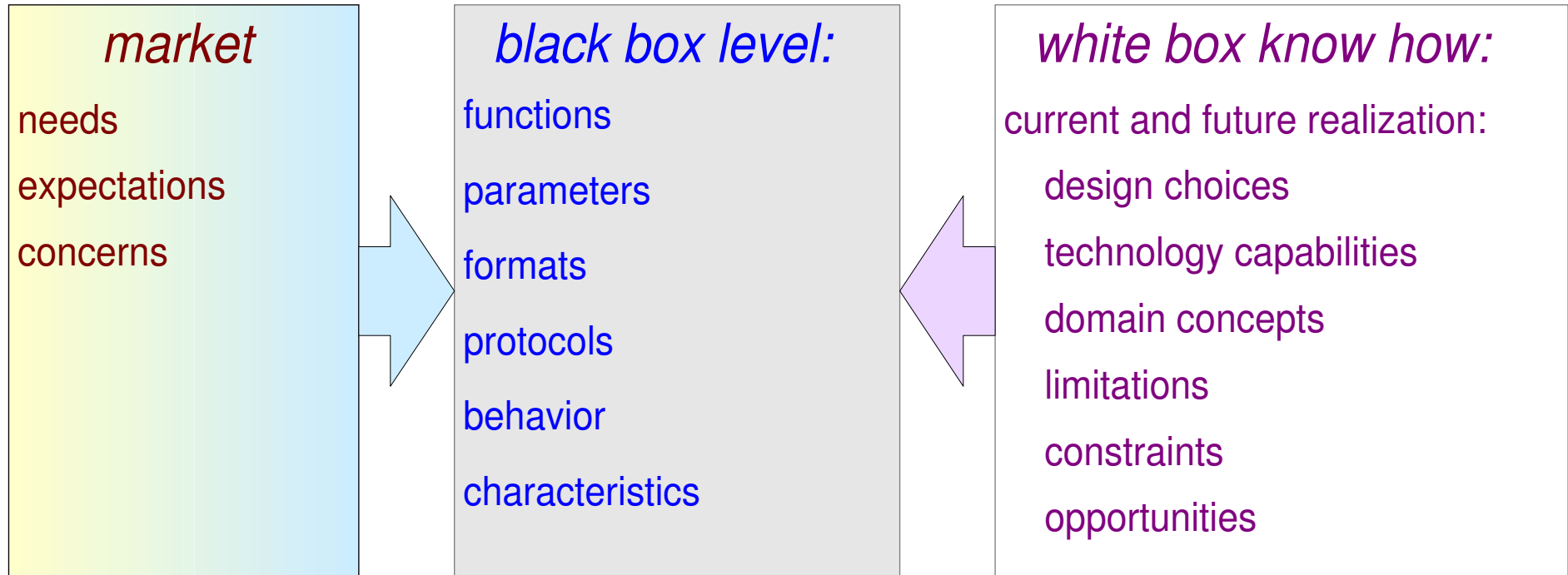
realism/acceptance level

time

effort

cost

Towards a Standard



future proof; room for innovation

market enabler; room for added value

not locked into specific technology constraints

realistic and acceptable; time, cost, effort

Standard: what

requirements at conceptual level,

no design or implementation

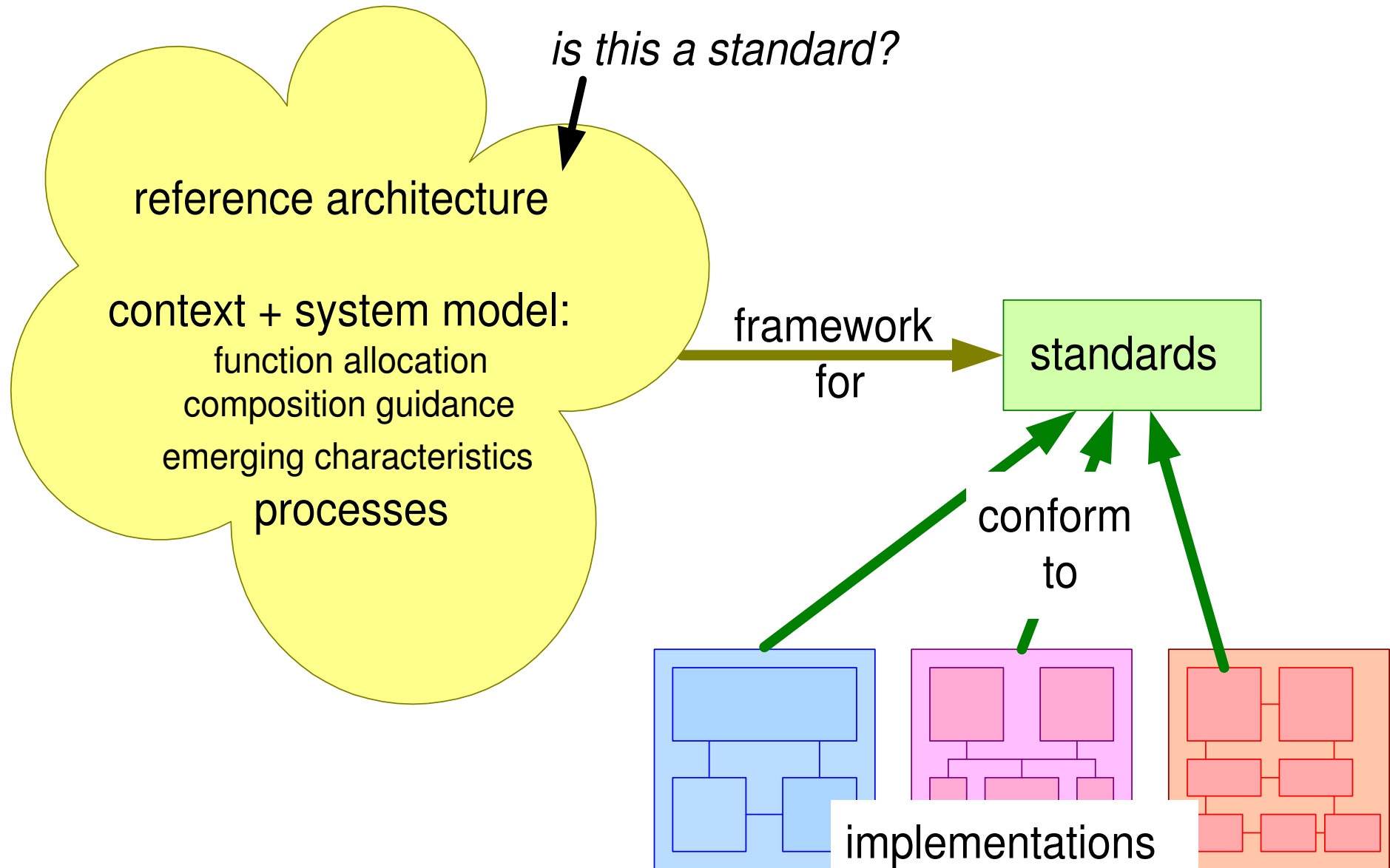
as minimal as possible

the minimal set of (interface) requirements to:

- 1) ensure interoperability
- 2) foster innovation and
- 3) maximise the room for added value.

ambitious but cautious

Embedding in a Reference Architecture



How to survive in
innovative domains?

standardization

what

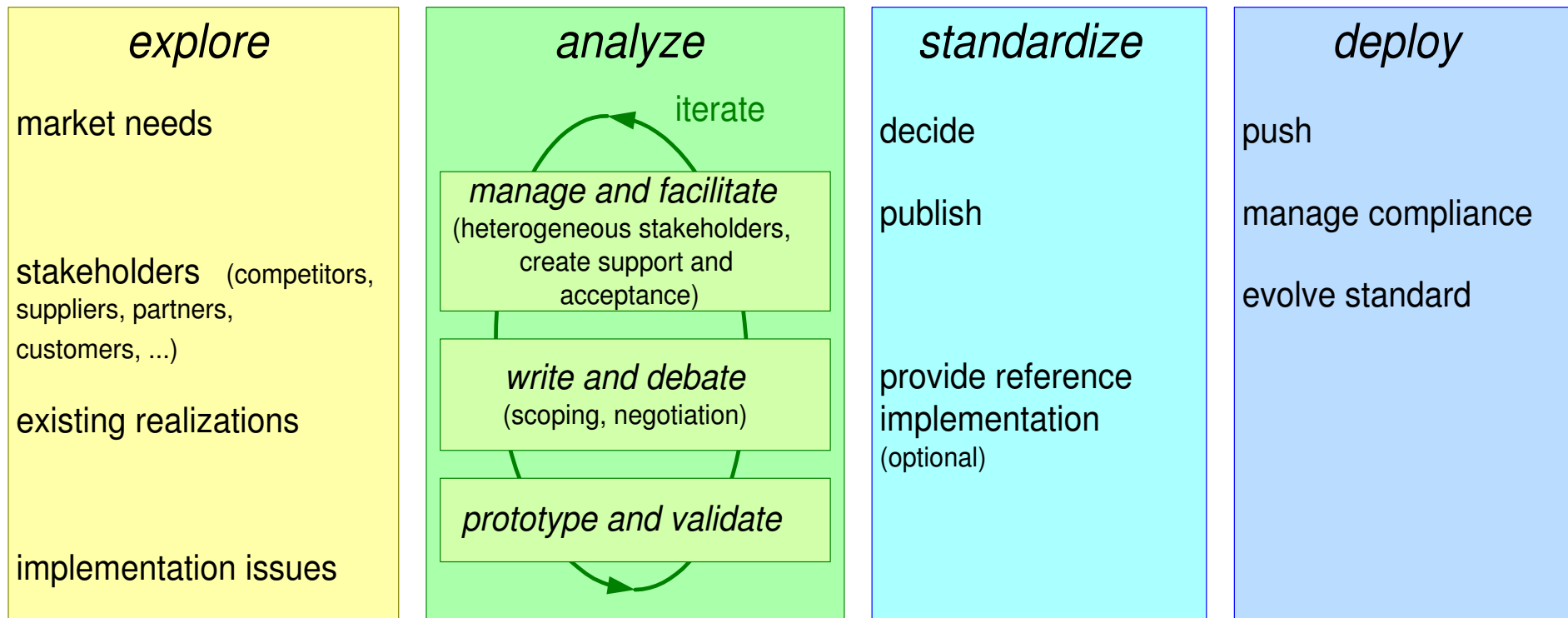
why

how

when

who

Flow of Standardization



Who Contributes and Participates?

How to survive in innovative domains?

standardization

what

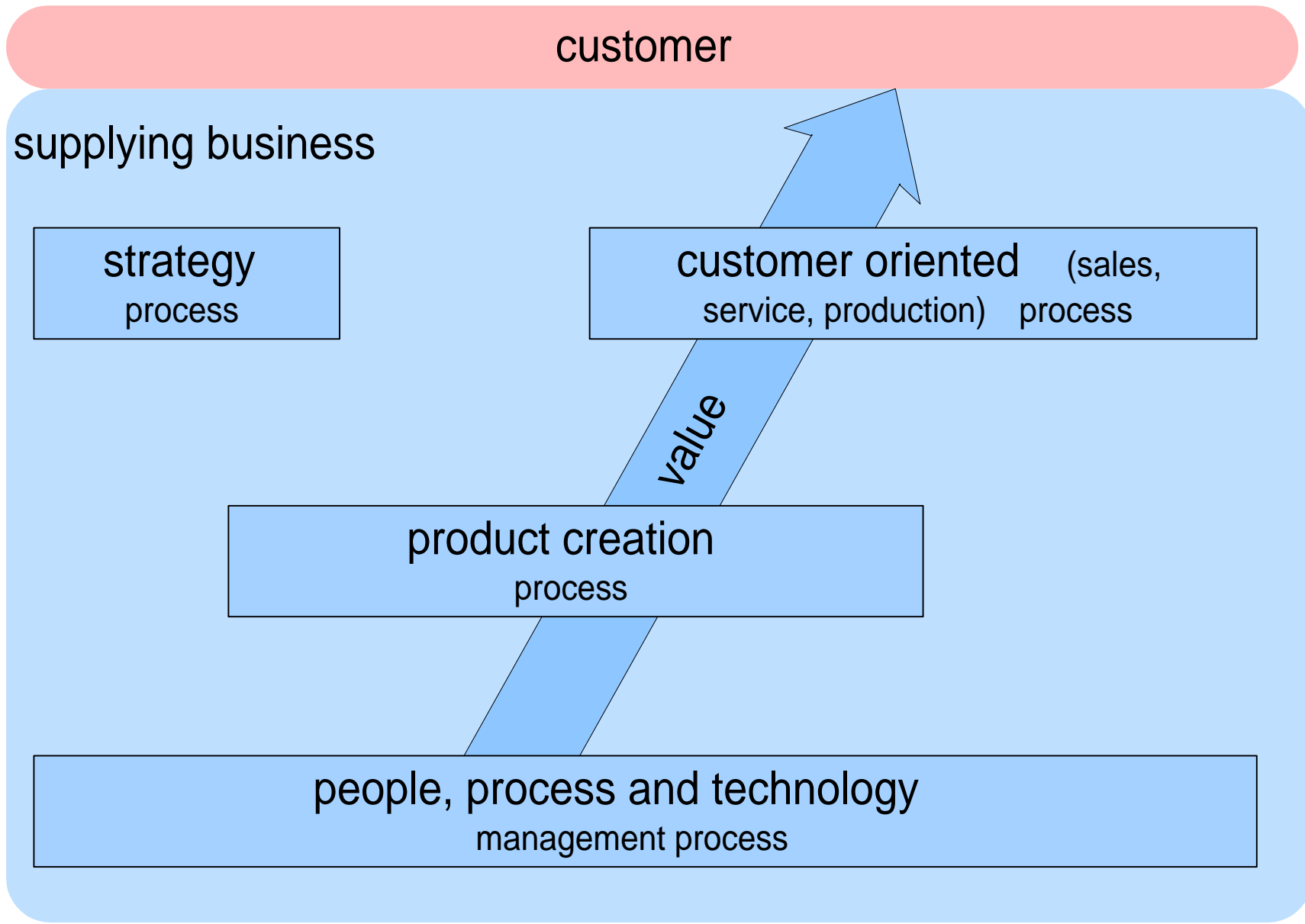
why

how

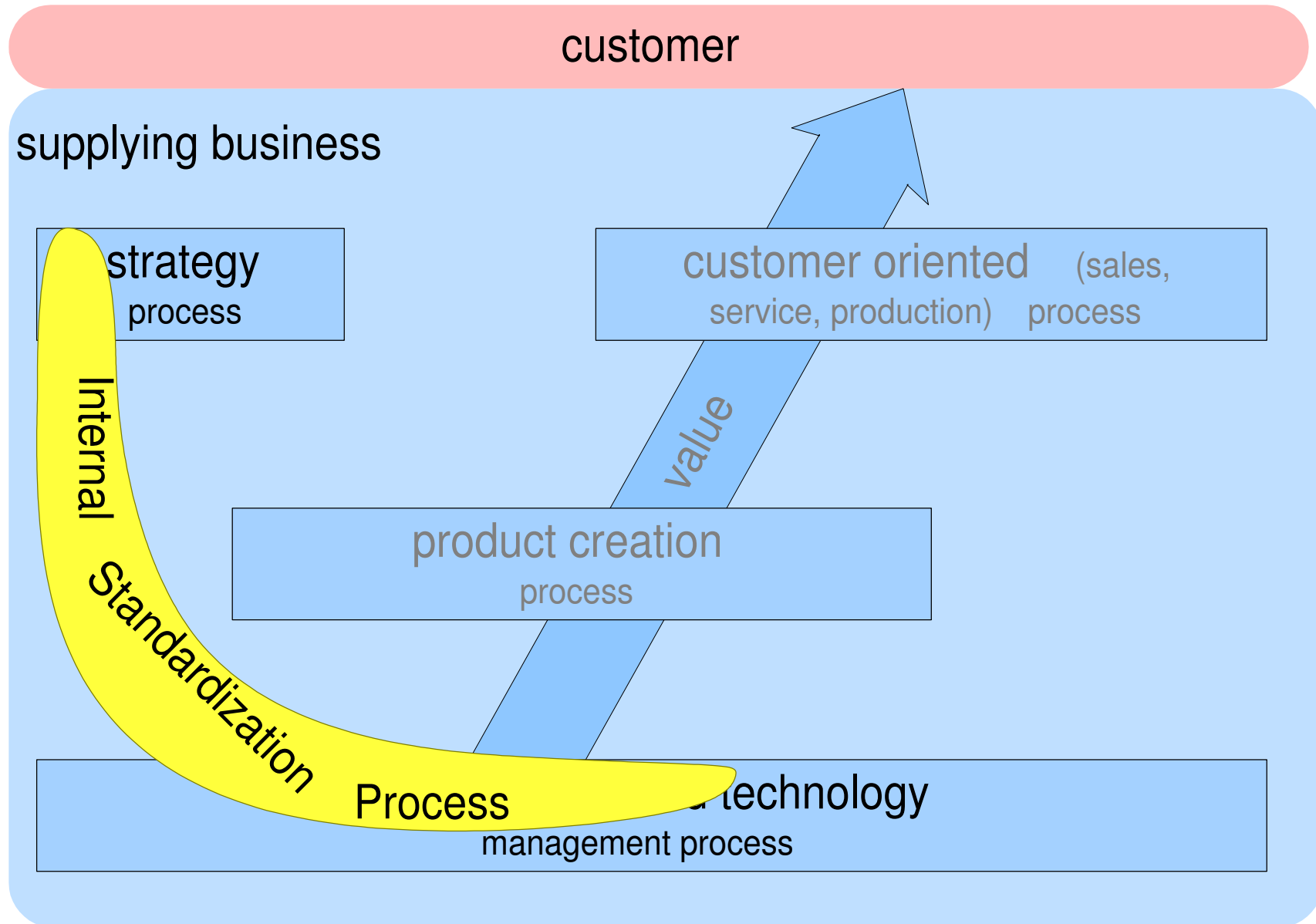
when

who

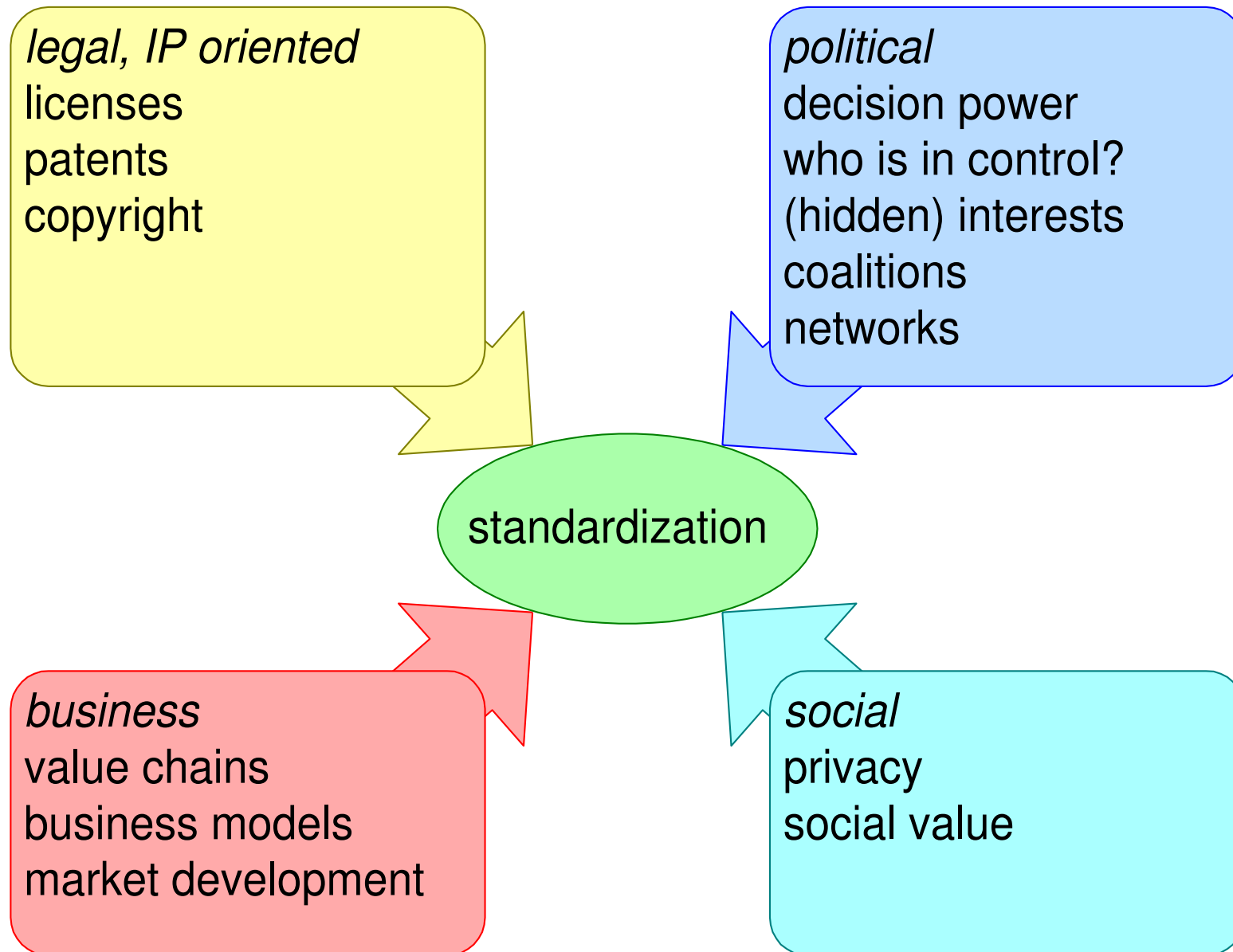
Simplified Process Decomposition



Internal Standardization Process == Highly Strategic!



Non technical aspects of standardization



Architect and Standards: Love-Hate Relationship

love

no worries: concerns are taken care of
focus on core problems
facilitates interoperability

hate

limits innovation (harness)
limits solution space
simplistic management orders

Conclusions

How to survive in innovative domains?

3. determine the right subjects and moments for *standardization*
4. apply a sensible *standardization* process

standardization

what

minimal, as little as possible requirements (not design or implementation)
room for added value and innovation

how

fast iteration
make rationale explicit
roadmapping

who

strategic insight
technology know how
market know how
social and political insight
ambitious but cautious

why

unlock market (e.g. interoperability)
focus on core assets
optimize supply chain

when

problem is understood
domain structure is clear
broadening set of stakeholders
technology is ripe