

Multi-view Architecting

by *Gerrit Muller, JürgenMüller, Jan Gerben Wijnstra*
Philips Research

e-mail: `gaudisite@gmail.com`

`www.gaudisite.nl`

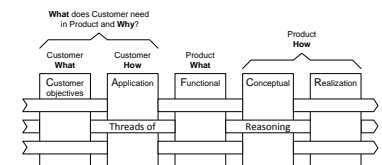
Buskerud University College,

Abstract

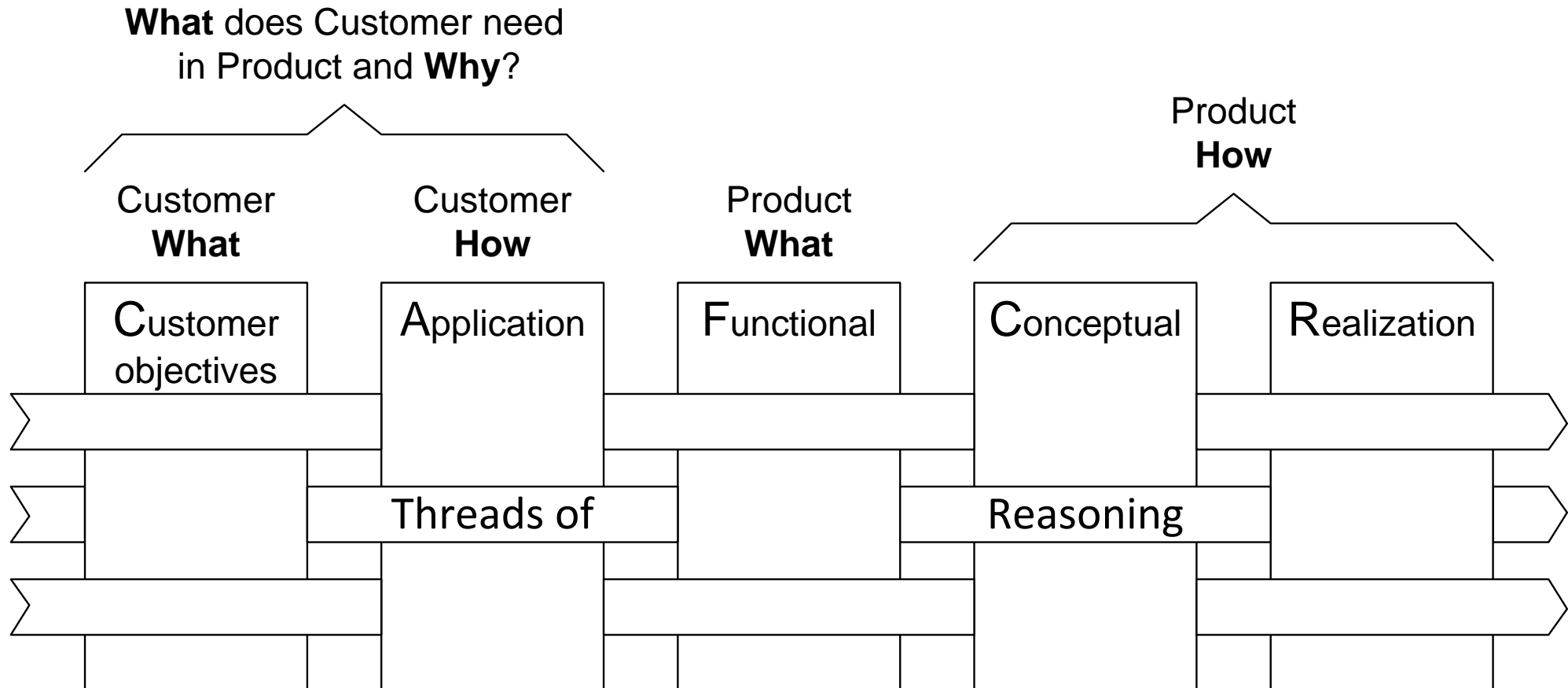
The development of large SW-intensive products needs to take requirements of multiple stakeholders into account. A design of such a system has to address functional and quality requirements adequately. However, for most of the required qualities no straight-forward design method exists even for a single quality.

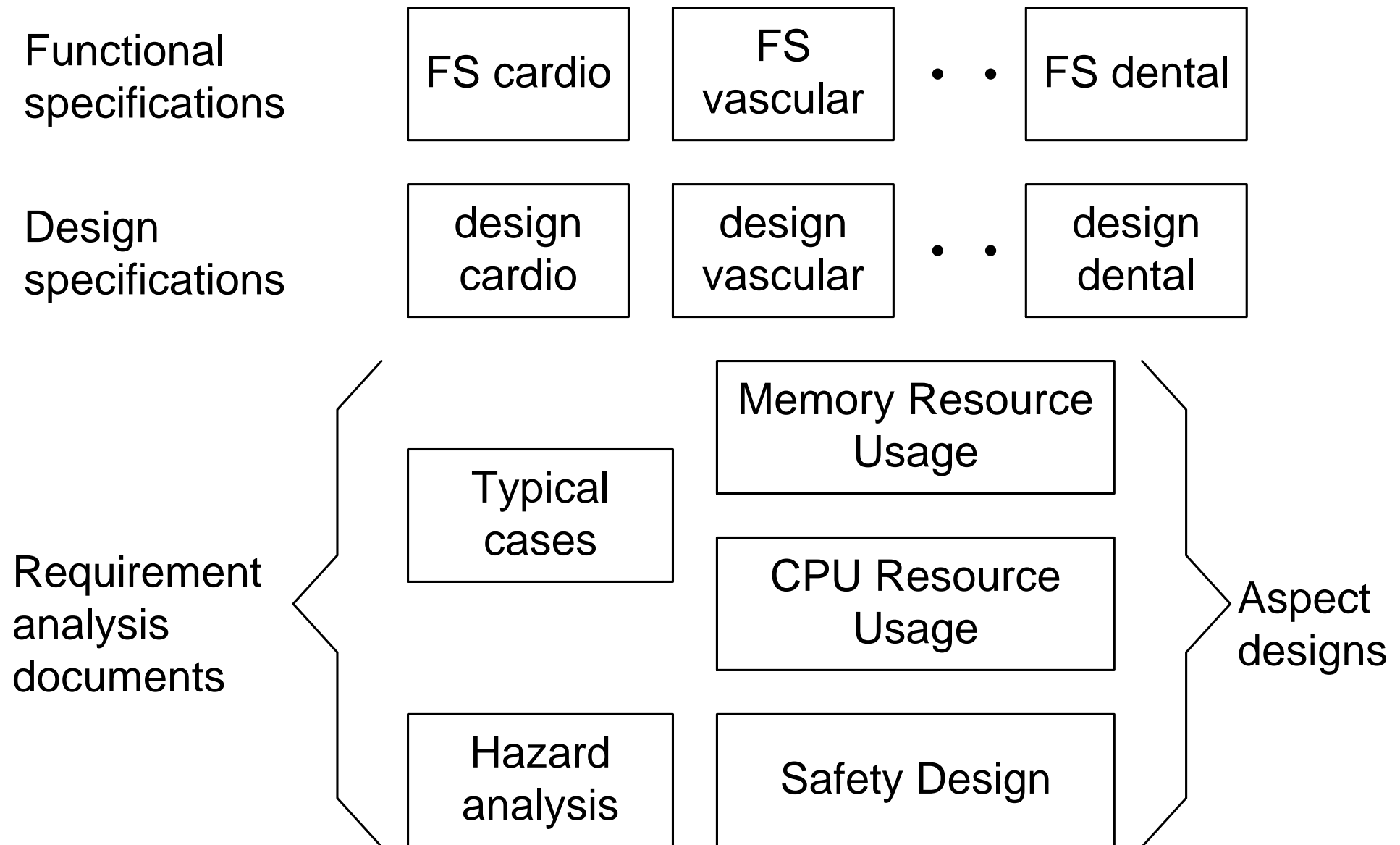
A multi-view architecting model is described based upon a decomposition of an architecture in 5 architectural views, ranging from customer objectives to realization. It is the task of the architect to keep these views consistent and to balance design decisions in the perspective of the stakeholder needs.

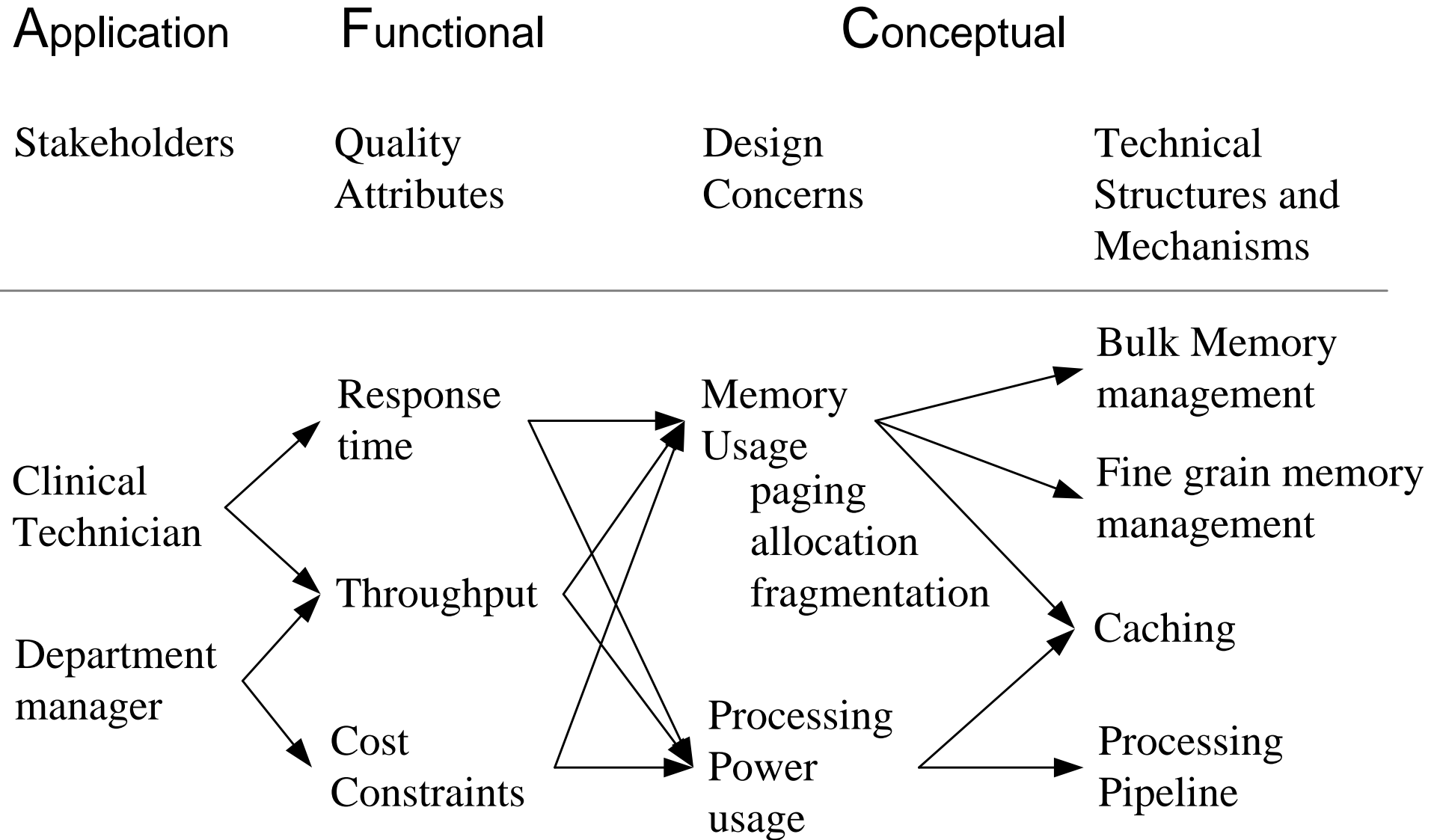
We derived this model from our experience in developing software intensive industrial products, 2 cases are described from the medical domain.



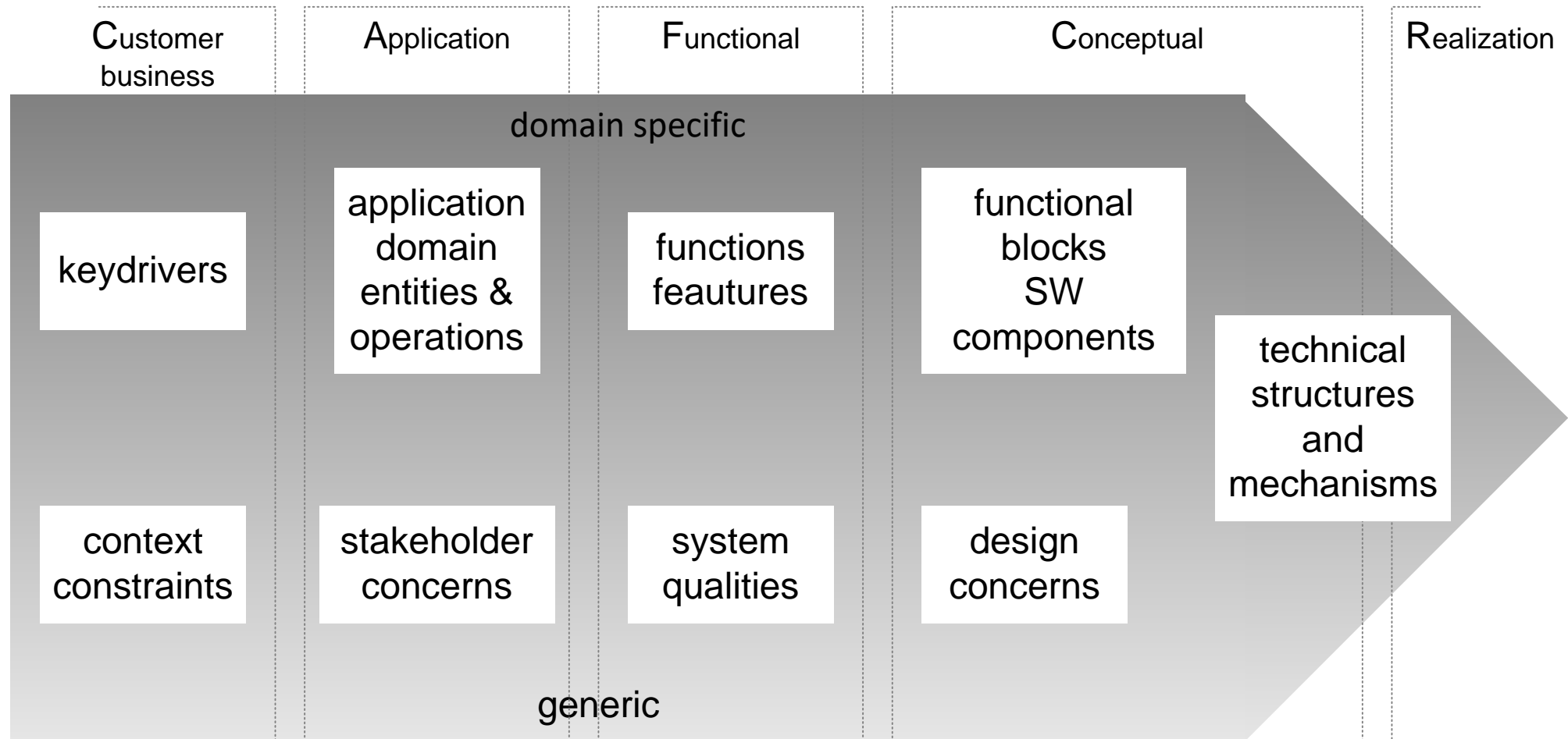
Integrating 5 System Architecture Views



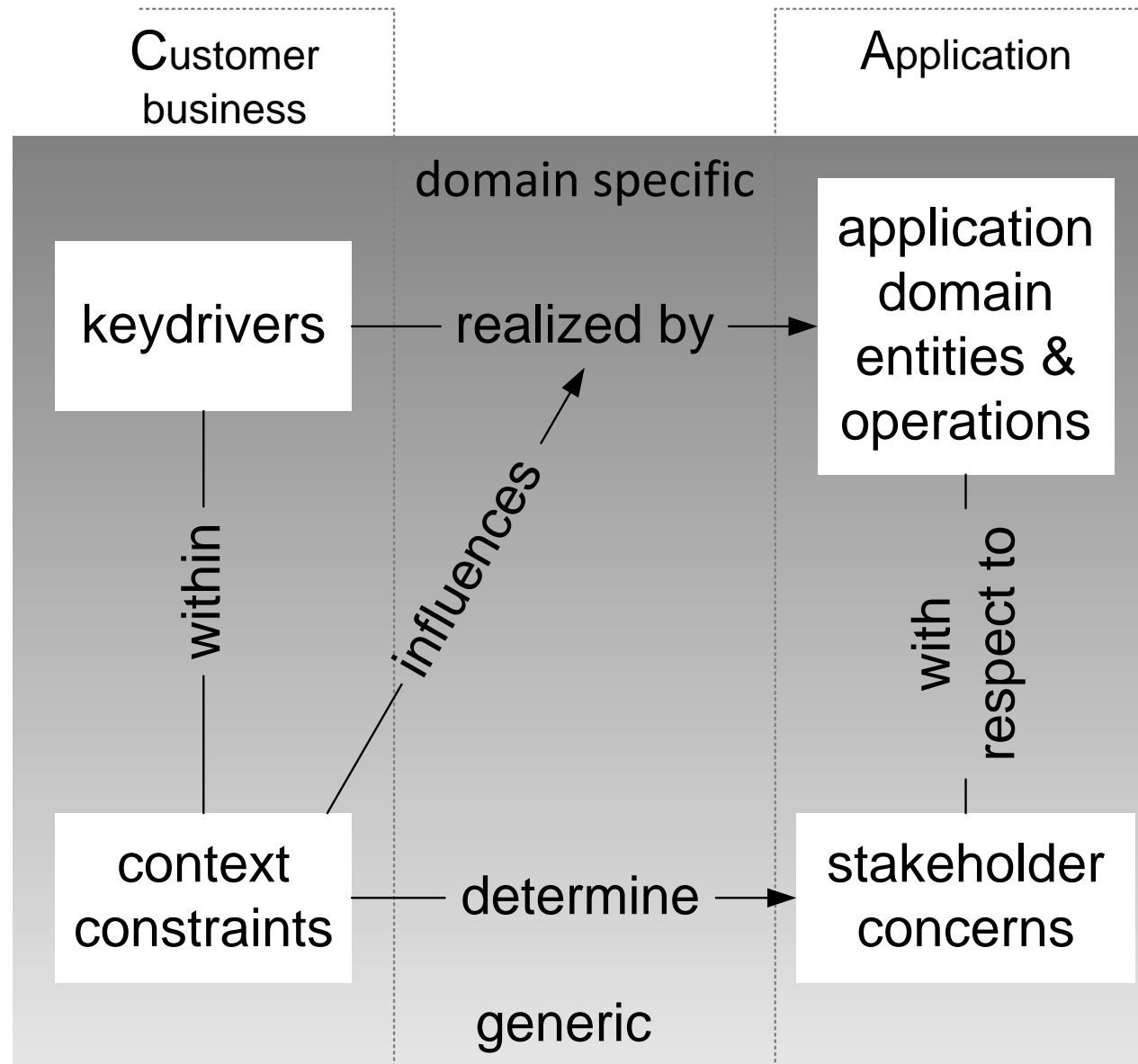




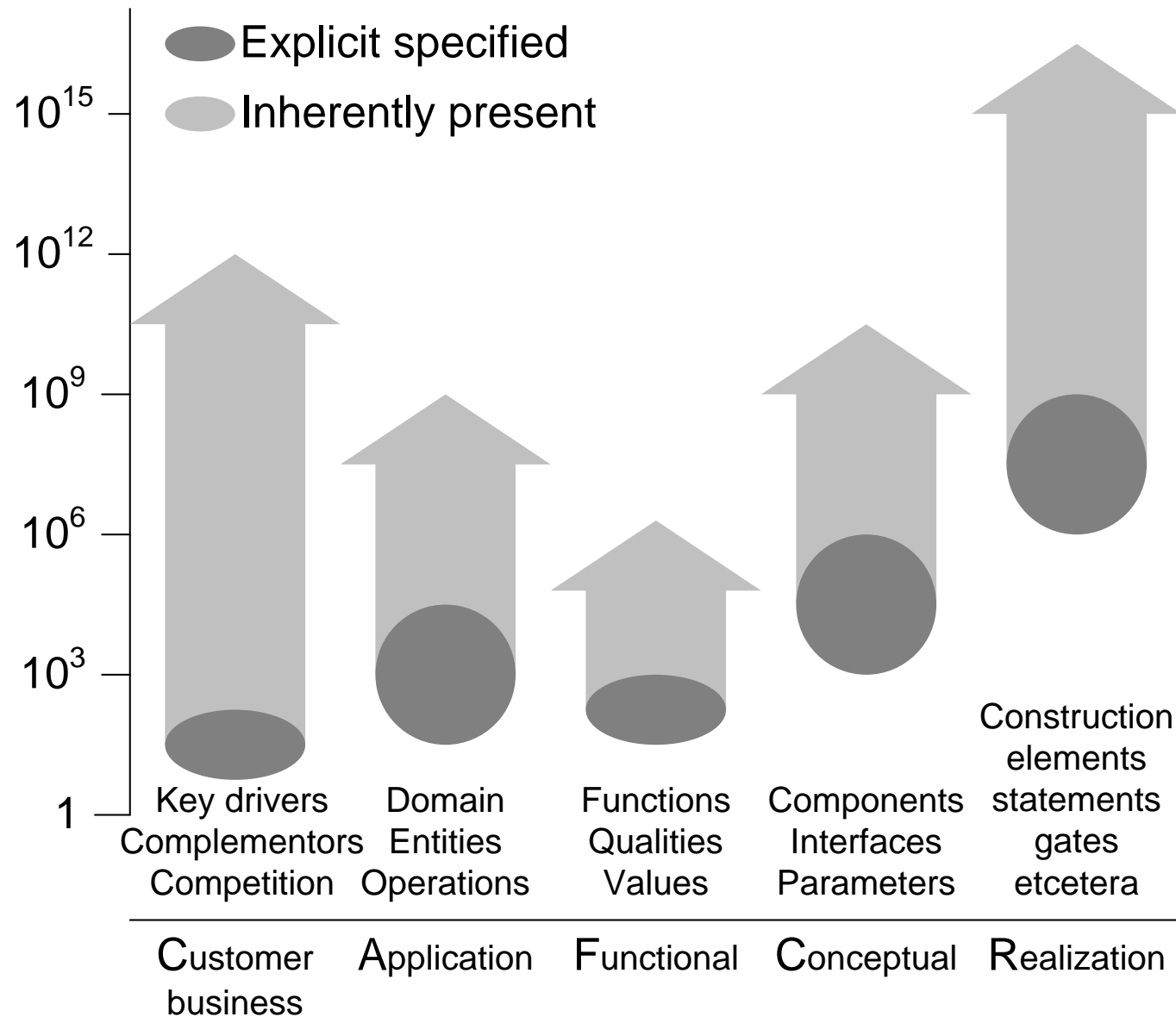
Issues per view



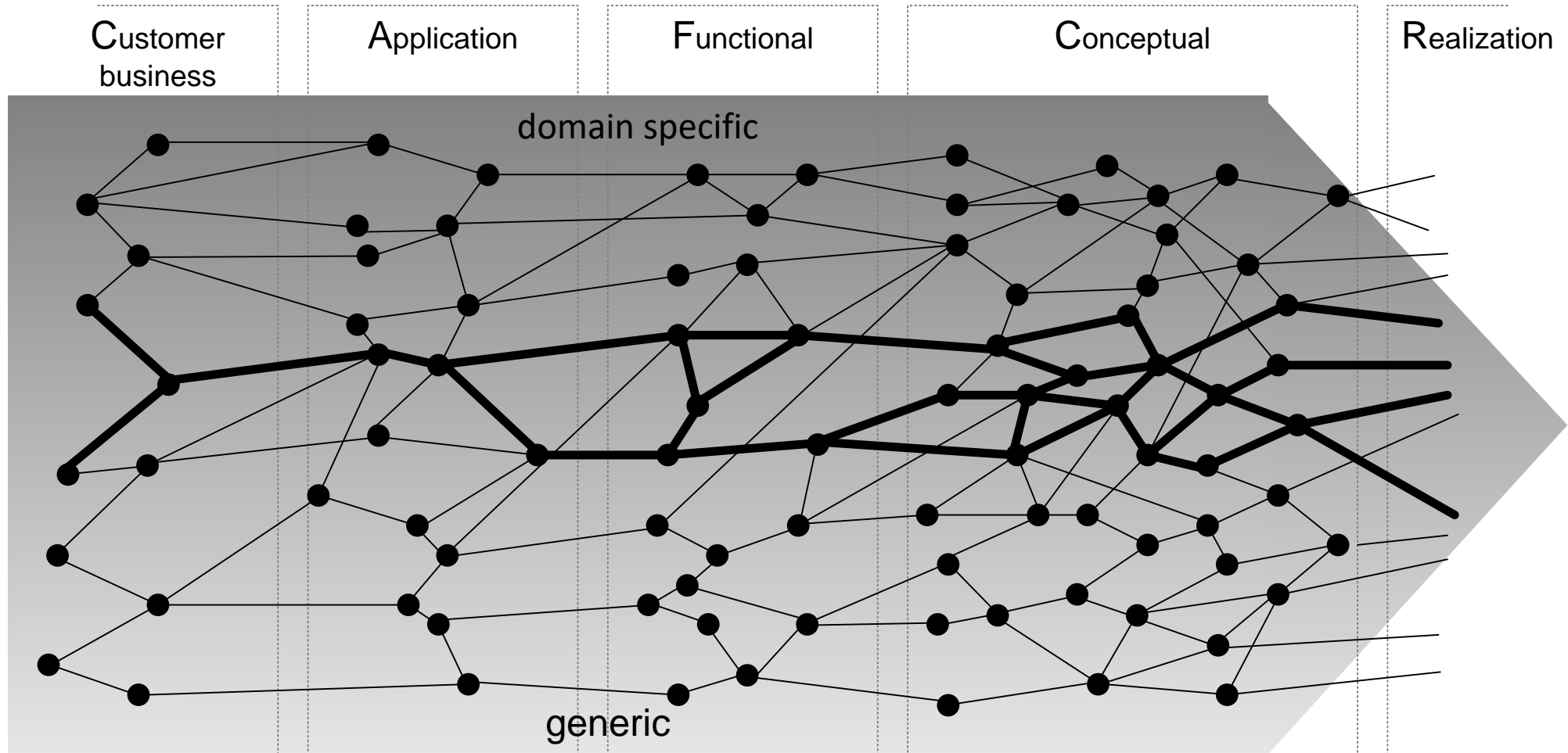
Zooming in on relations



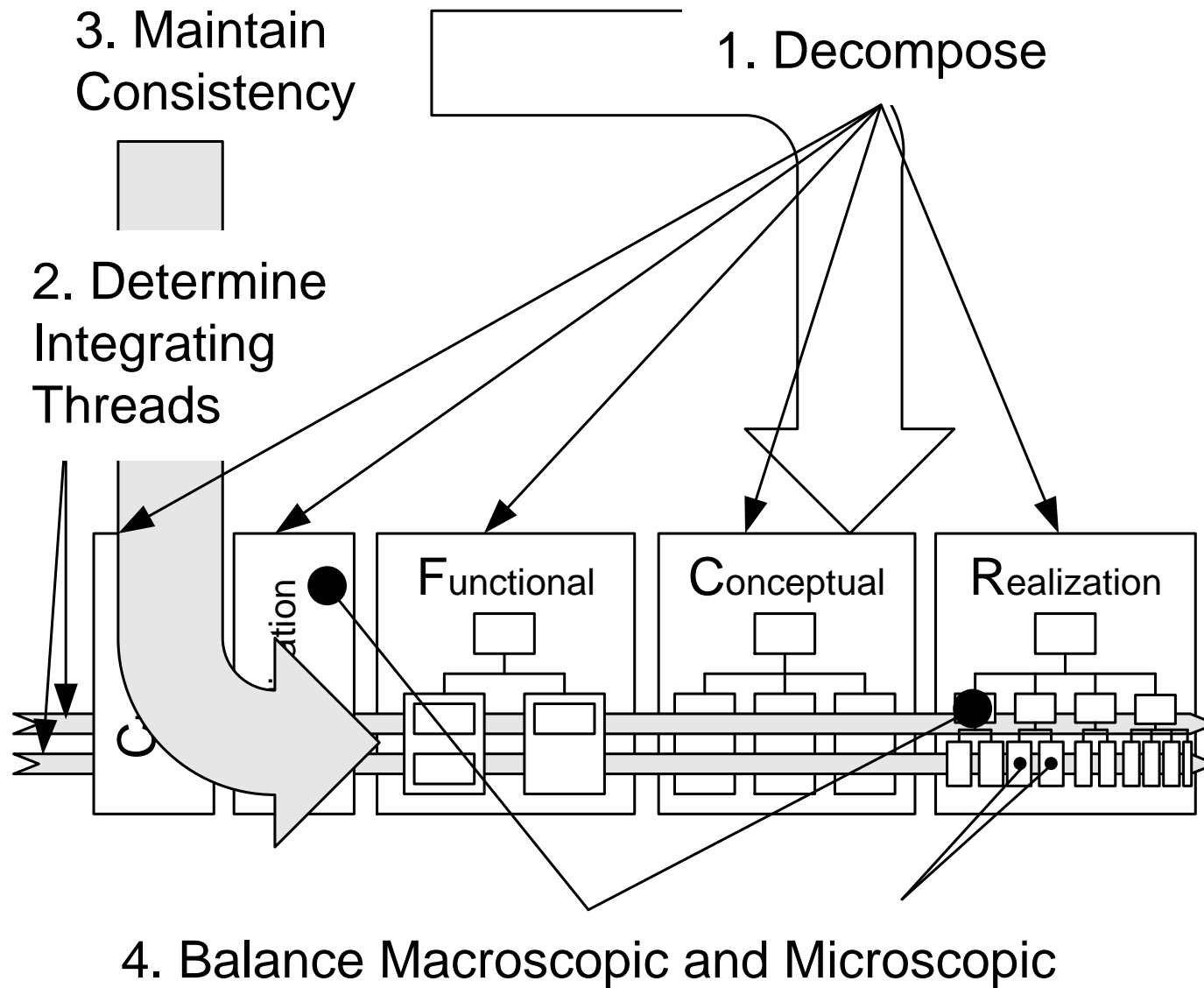
Explicit facts and inherent details per view



One thread of reasoning



Activities in multi-view architecting



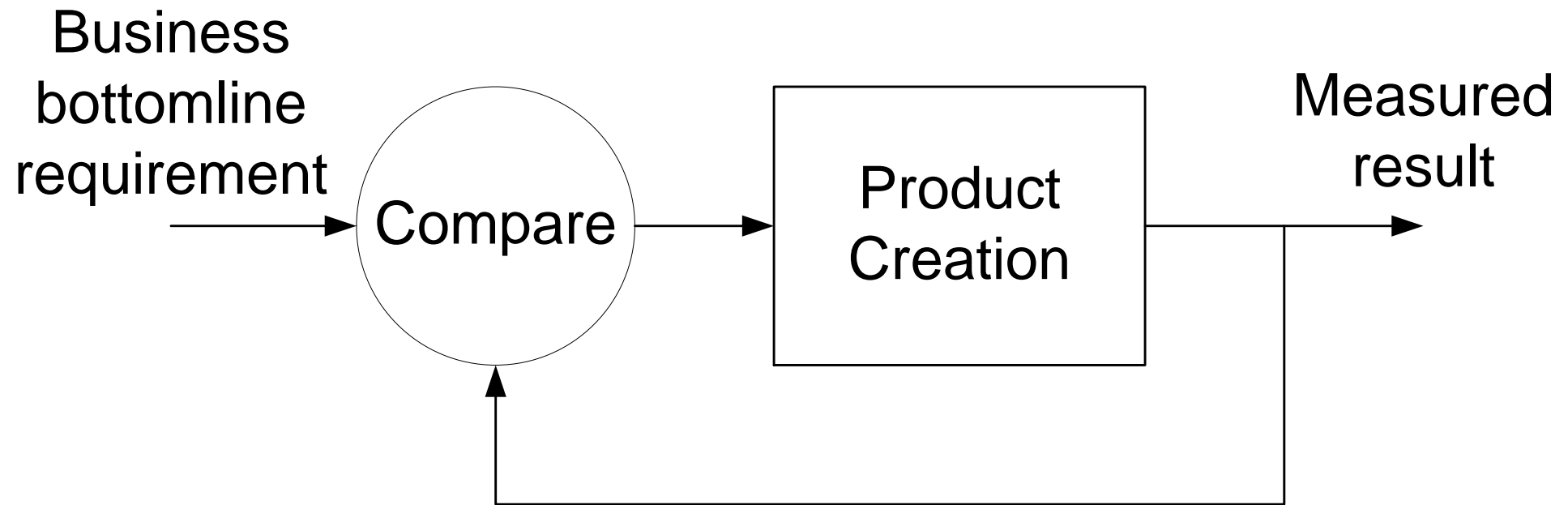
Criteria for thread selection

- Important for customer and the business
- Critical with respect to technical realization

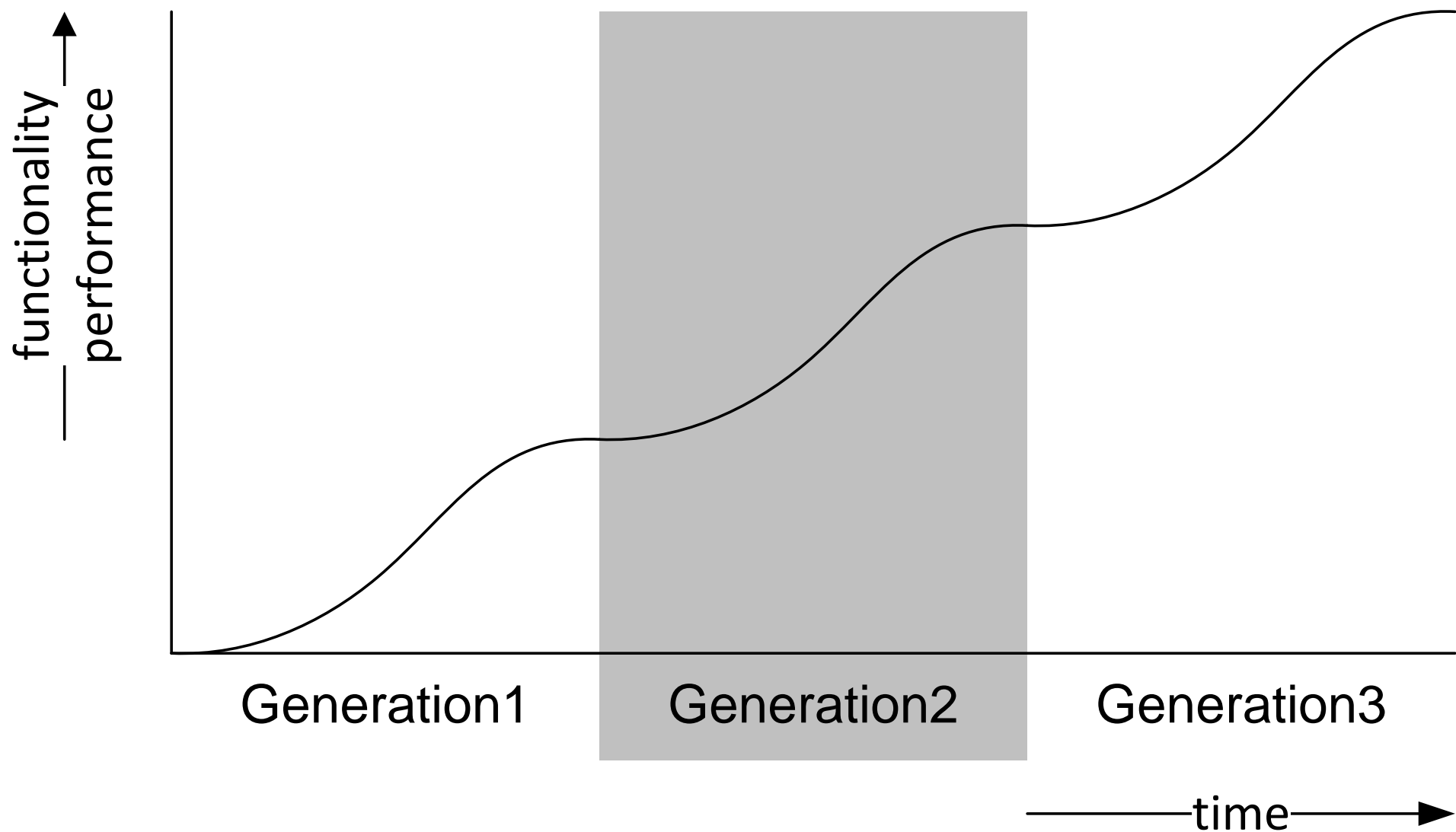
Pitfalls in multi-view architecting

- too few views
- completeness
- general formalization

Continuous feedback during Product Creation



Stepwise evolution



Qualities checklist

Safety	Manufacturability	Cost price
Security	Testability	Cost of operation
Reliability	Serviceability	Interaction with environment
Robustness	Configurability	Power consumption
Useability	Installability	Consumption rate (water, air, chemicals, etcetera)
Appeal, Appearance	Evolvability	Disposability
Throughput or Productivity	Portability	Size, weight
Response Time	Upgradeability	Resource utilization
Image Quality	Extendability	
Reproduceability	Maintainability	
Predicatability	Logistics flexibility	
Accuracy	Lead time	
Transportability	Standards Compliance	
Wearability		
Storability		

SW aspects checklist

granularity

- scoping
- containment
- cohesion
- coupling

interfaces

allocation

- budgets

information model

- entities
- relations
- operations

characteristics

- static
- dynamic

configuration man.

- packages
- components
- files
- objects
- modules
- interfaces

meta-functional

- operational
 - image processing
 - handling calls

..

initialization

- start-up
- shutdown
- bootstrap
- discovery
- negotiation

fault handling

- exceptions
- logs
- traces

diagnostics

- configuration handling
- data replication
- performance observation
- capability query

testing

- automation
- special methods
- harness
- suites

off-line guidance

supply chain

- outsource
- co-design
- buy
- interoperate
- source vs binary

technology choices

- lifecycle
- obsolescence
- core, key, base

SW development

- environment
- repository
- tools

feedback tools

- monitoring
- statistics
- analysis
- call graphs
- message tracing
- object tracing

licensing

- SW keys

synchronization identification

- signalling
- messaging
- call-back scheduling
- notification
- active data
- watchdogs
- time-outs
- locking
- semaphores
- transactions
- checkpoints
- deadlock detection
- roll-back
- priorities
- pre-emption

concurrency

- processes
- tasks
- threads

persistence

- caching
- versioning,
- prefetching
- lazy evaluation

- uniqueness
- naming
- data model,
- registry
- scoping
- configuration
- database
- inheritance

resource management

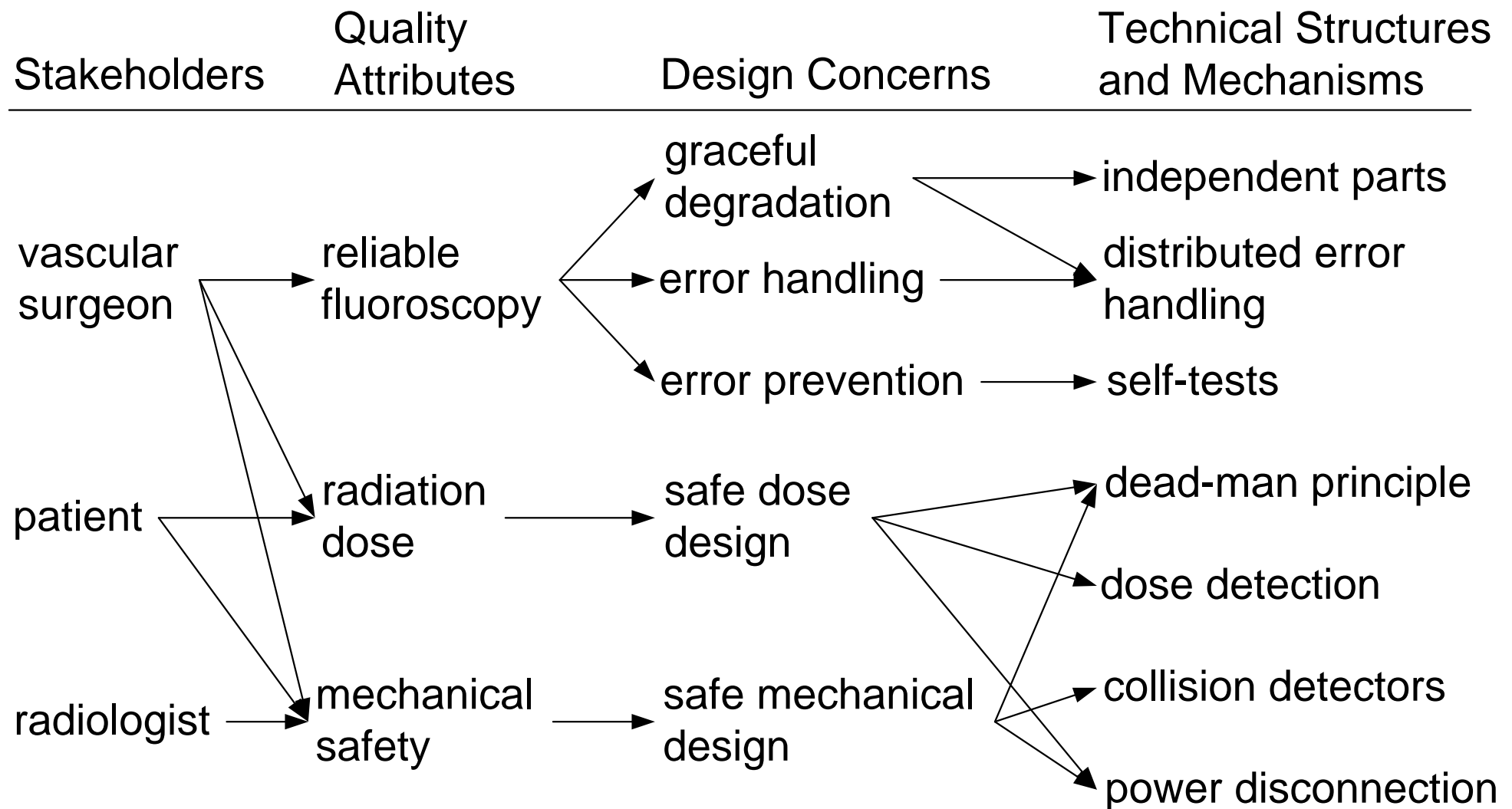
- allocation
- anti-fragmentation
- garbage collection

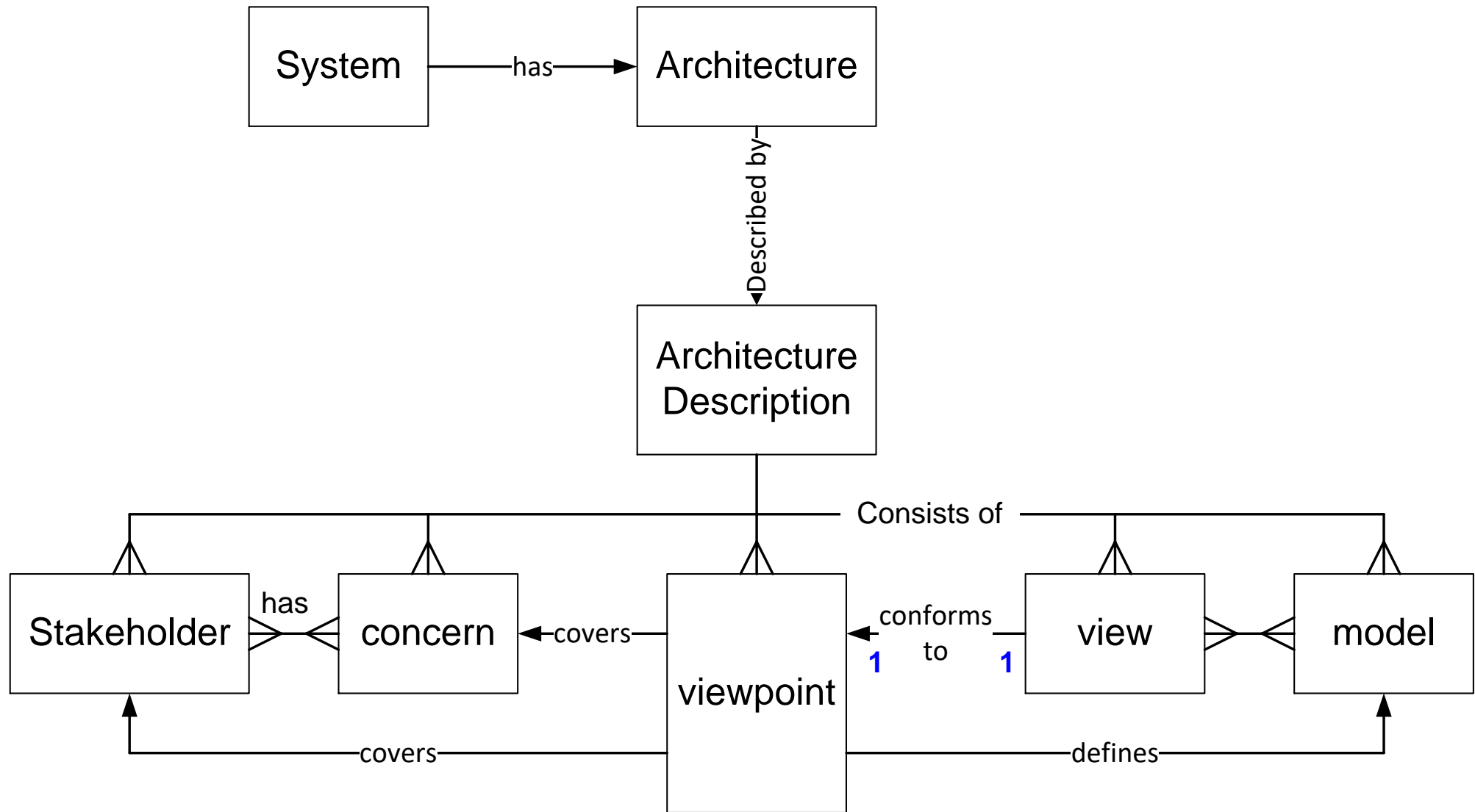
distribution

- allocation,
- transparency
- component,
- client/Server
- multi-tier model

infrastructure

Medical Safety





ISO 9126 quality framework

- **Functionality** suitability, accuracy, interoperability, compliance, security, *traceability*
- **Reliability** maturity, fault tolerance, recoverability, *availability, degradability*
- **Usability** understandability, learnability, operability, *explicitness, customisability, attractiveness, clarity, helpfulness, user-friendliness*
- **Efficiency** time behaviour, resource behaviour
- **Maintainability**
- **Portability**

