Architectural documentation, views and tradeoffs, patterns

Claire Le Goues

October 3, 2019

Administrativia

Learning Goals

- Distinguish software architecture from (object-oriented) software design
- Use notation and views to describe the architecture suitable to the purpose, and document architectures clearly and without ambiguity.
- Use diagrams to understand systems and reason about tradeoffs.
- Understand the utility of architectural patterns and tactics, and give a couple of examples.

ARCHITECTURE VS OBJECT-LEVEL DESIGN

One slide on a thing from 214: Design

- Design process (analysis, design, implementation)
- Design goals (cohesion, coupling, information hiding, design for reuse, …)
- Design patterns (what they are, for what they are useful, how they are described)
- Frameworks and libraries (reuse strategies)

Levels of Abstraction

- Requirements
  - high-level “what” needs to be done
- Architecture (High-level design)
  - high-level “how”, mid-level “what”
- OO-Design (Low-level design, e.g. design patterns)
  - mid-level “how”, low-level “what”
- Code
  - low-level “how”

Design vs. Architecture

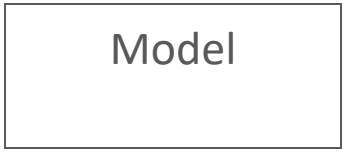
Design Questions

- How do I add a menu item in Eclipse?
- How can I make it easy to add menu items in Eclipse?
- What lock protects this data?
- How does Google rank pages?
- What encoder should I use for secure communication?
- What is the interface between objects?

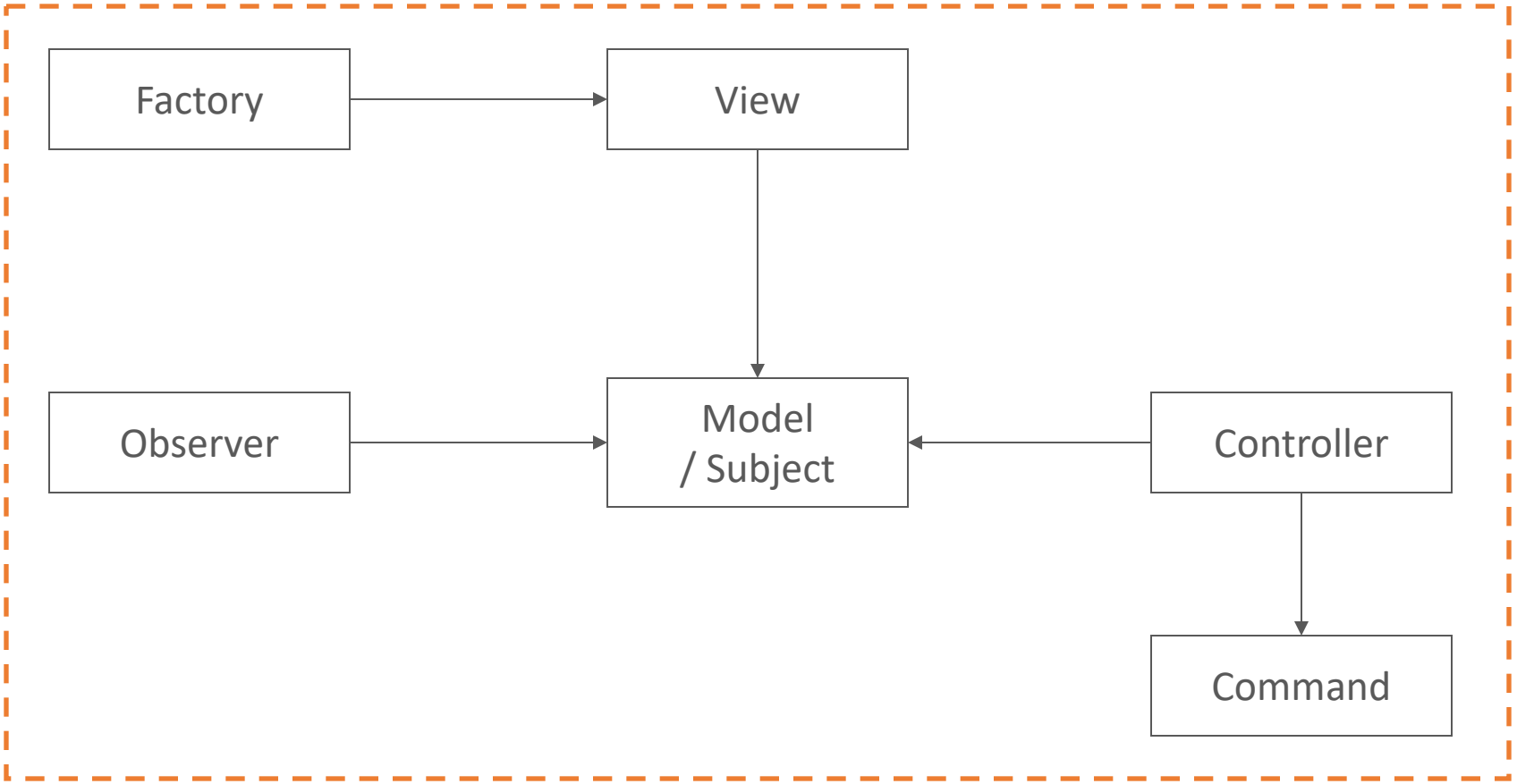
Architectural Questions

- How do I extend Eclipse with a plugin?
- What threads exist and how do they coordinate?
- How does Google scale to billions of hits per day?
- Where should I put my firewalls?
- What is the interface between subsystems?

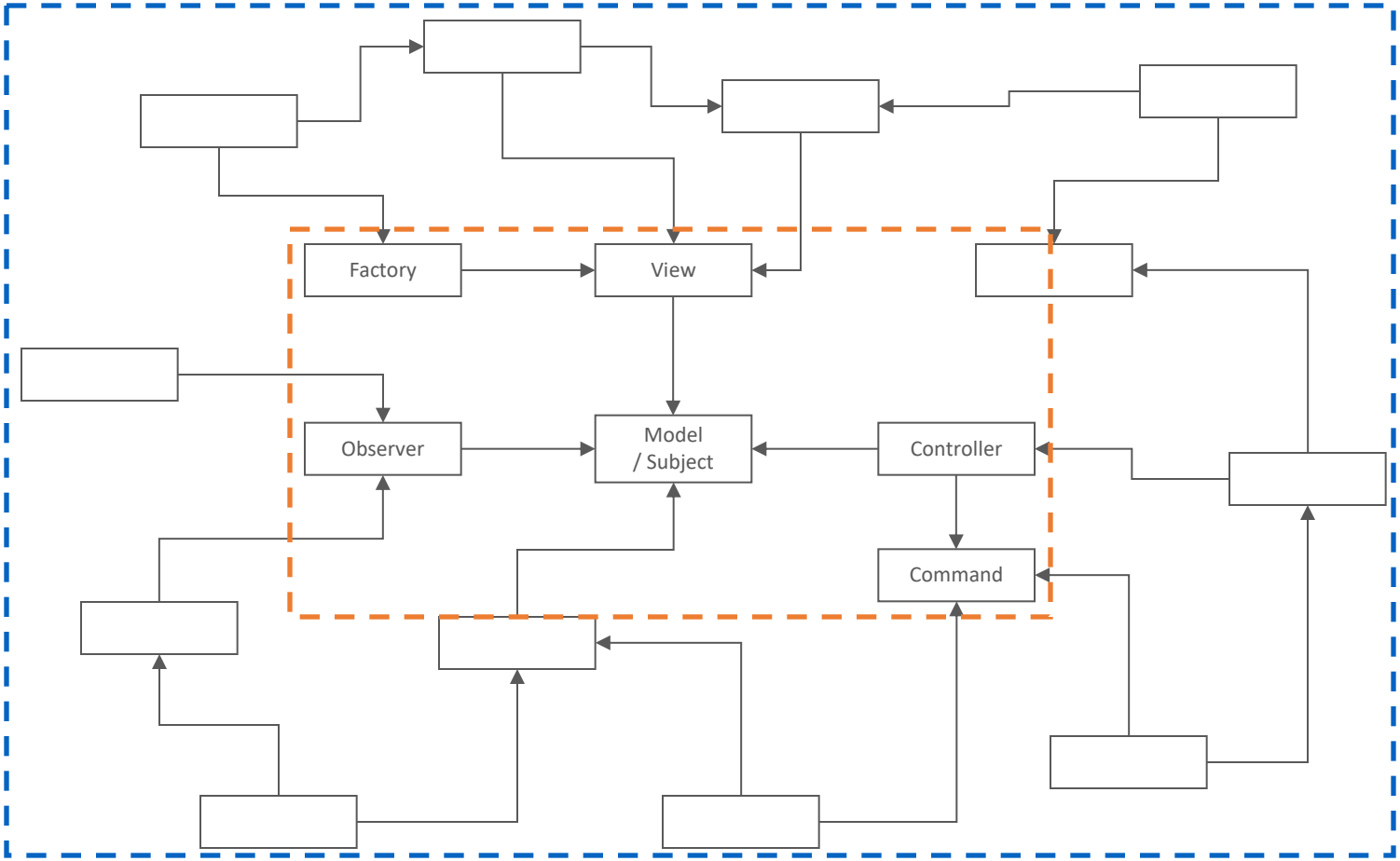
Objects



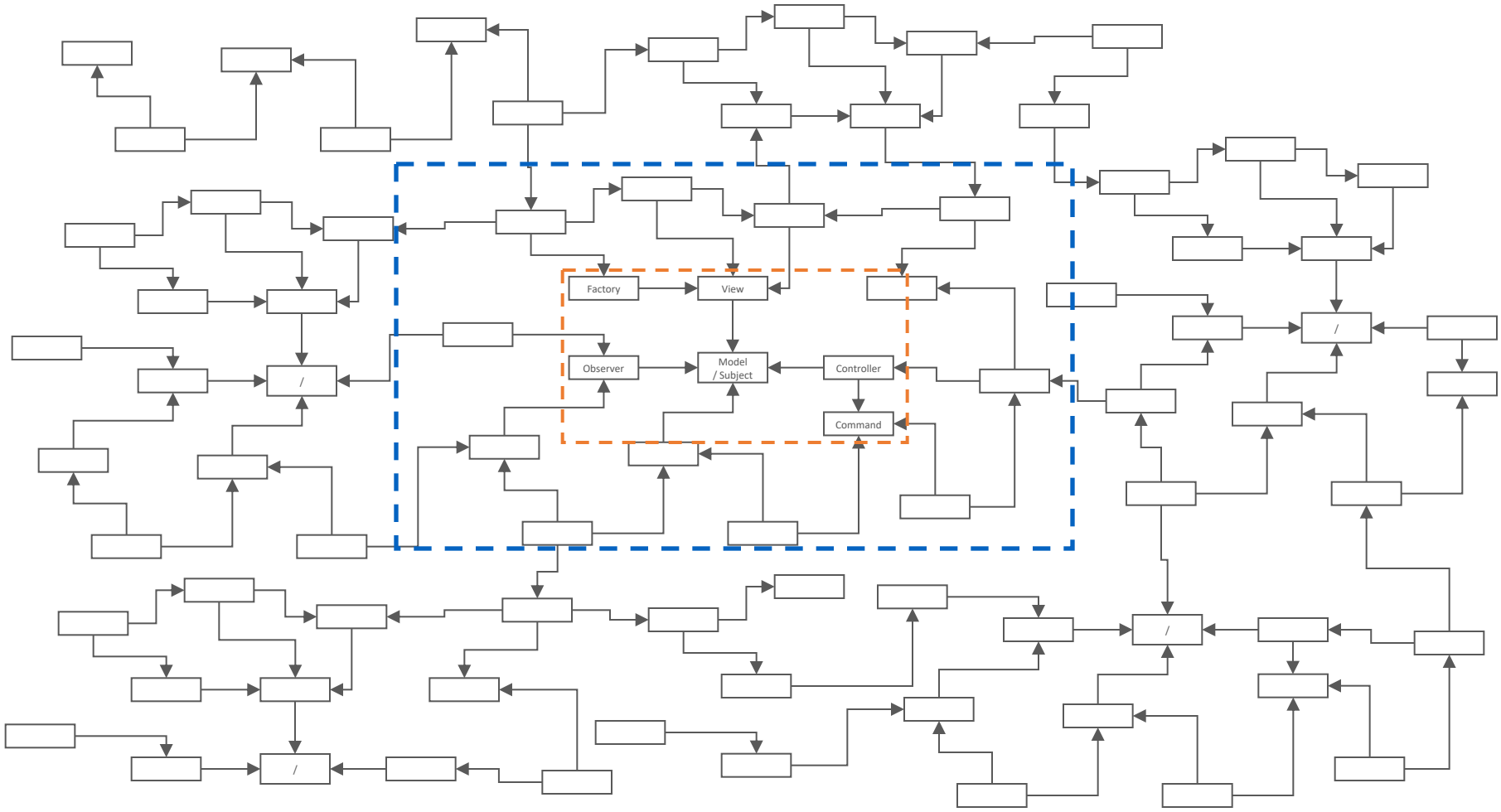
Design Patterns



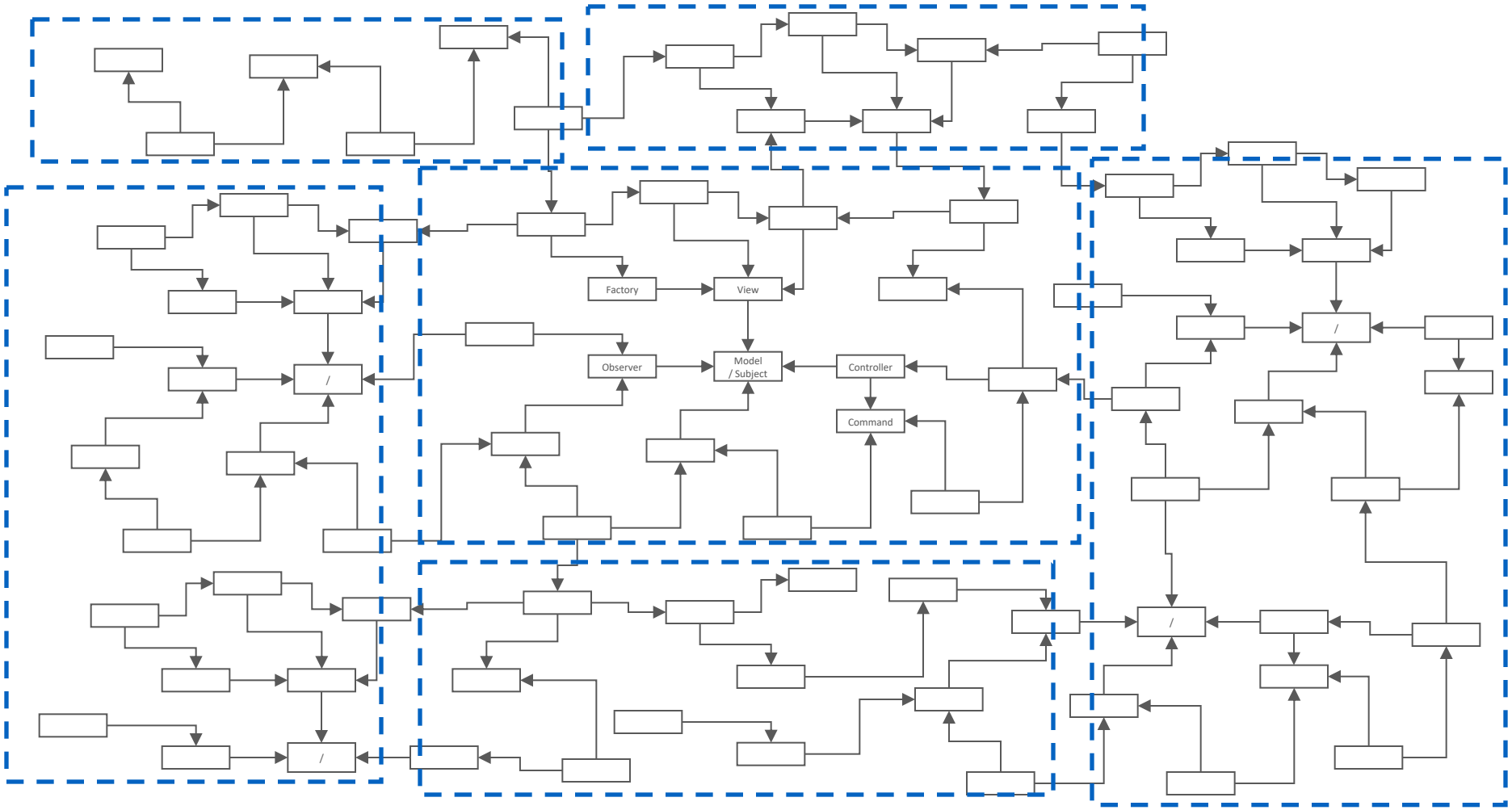
Design Patterns



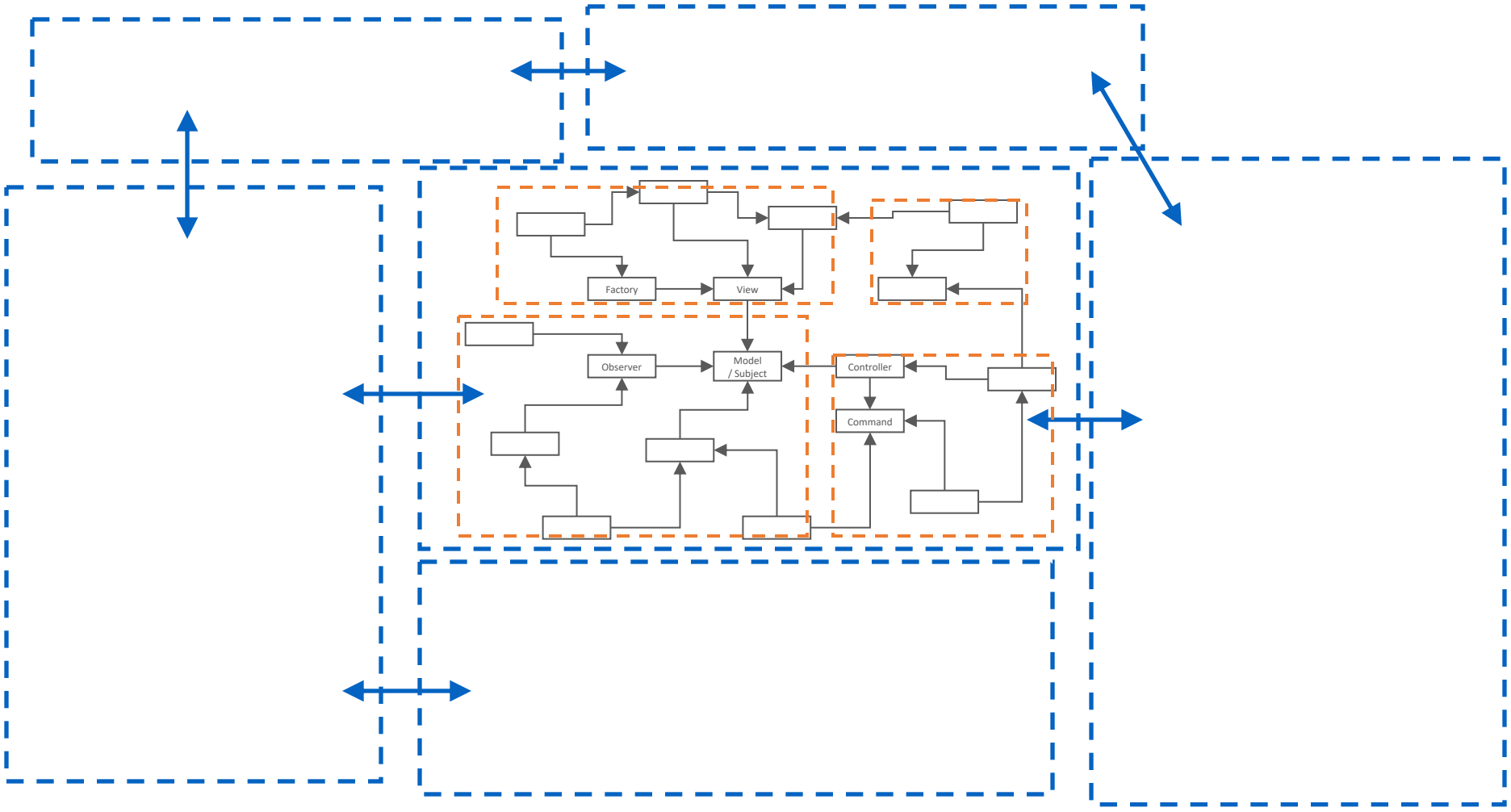
Design Patterns



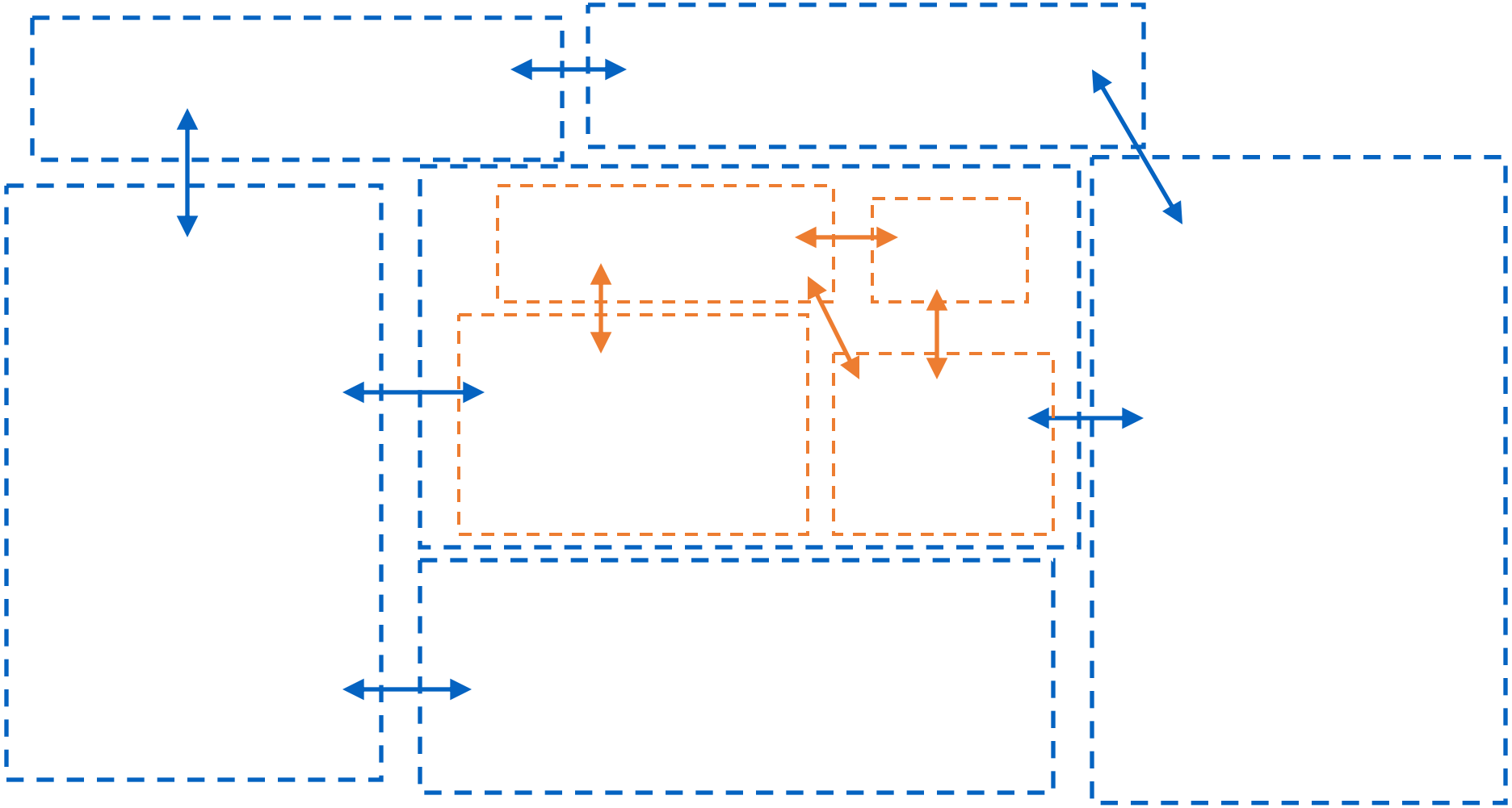
Architecture



Architecture



Architecture

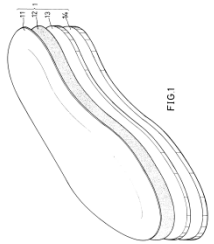


ARCHITECTURE DOCUMENTATION & VIEWS

Architecture Disentangled

Architecture as
structures and relations
(the actual system)

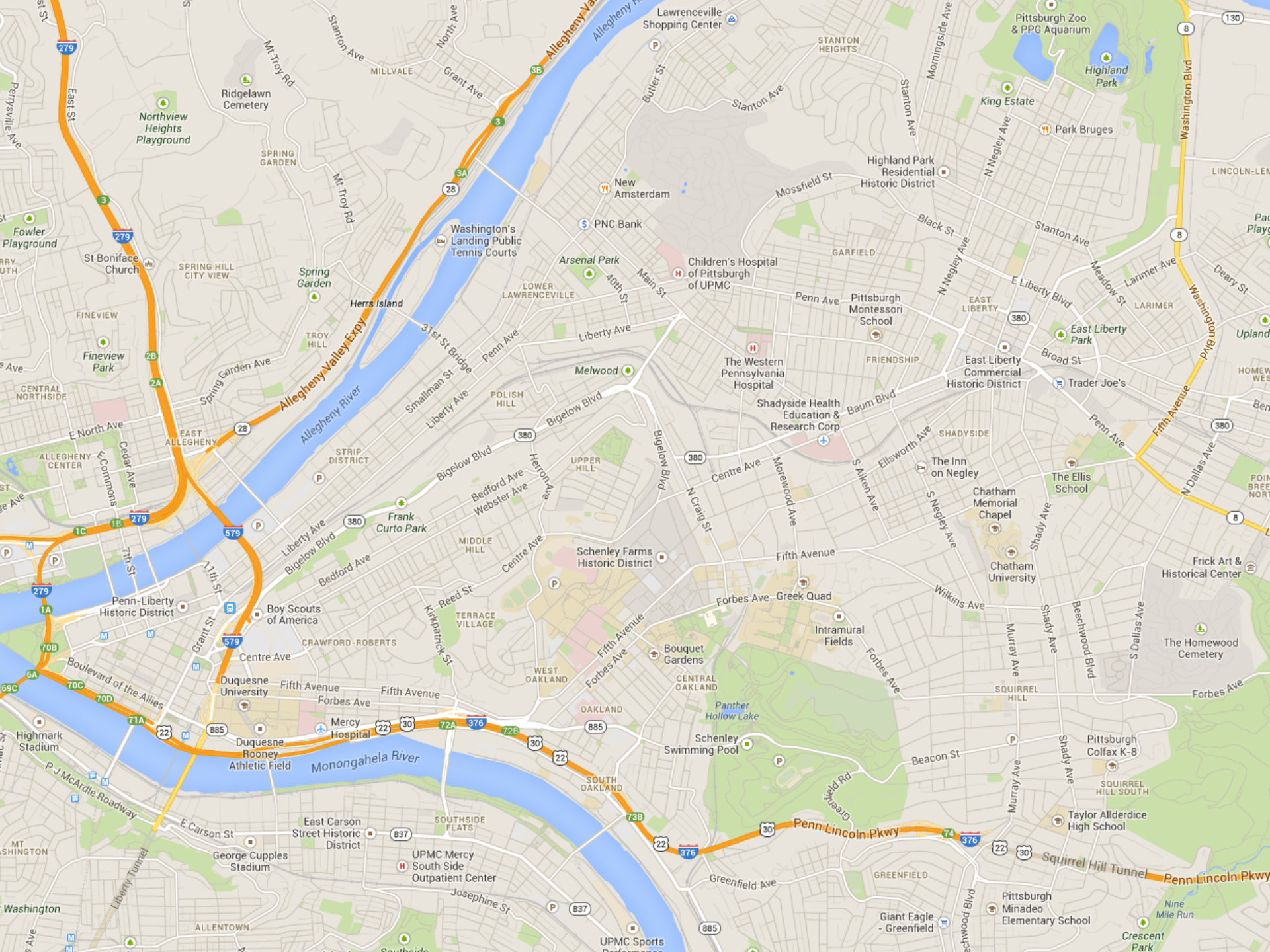
Architecture as
documentation
(representations of the system)

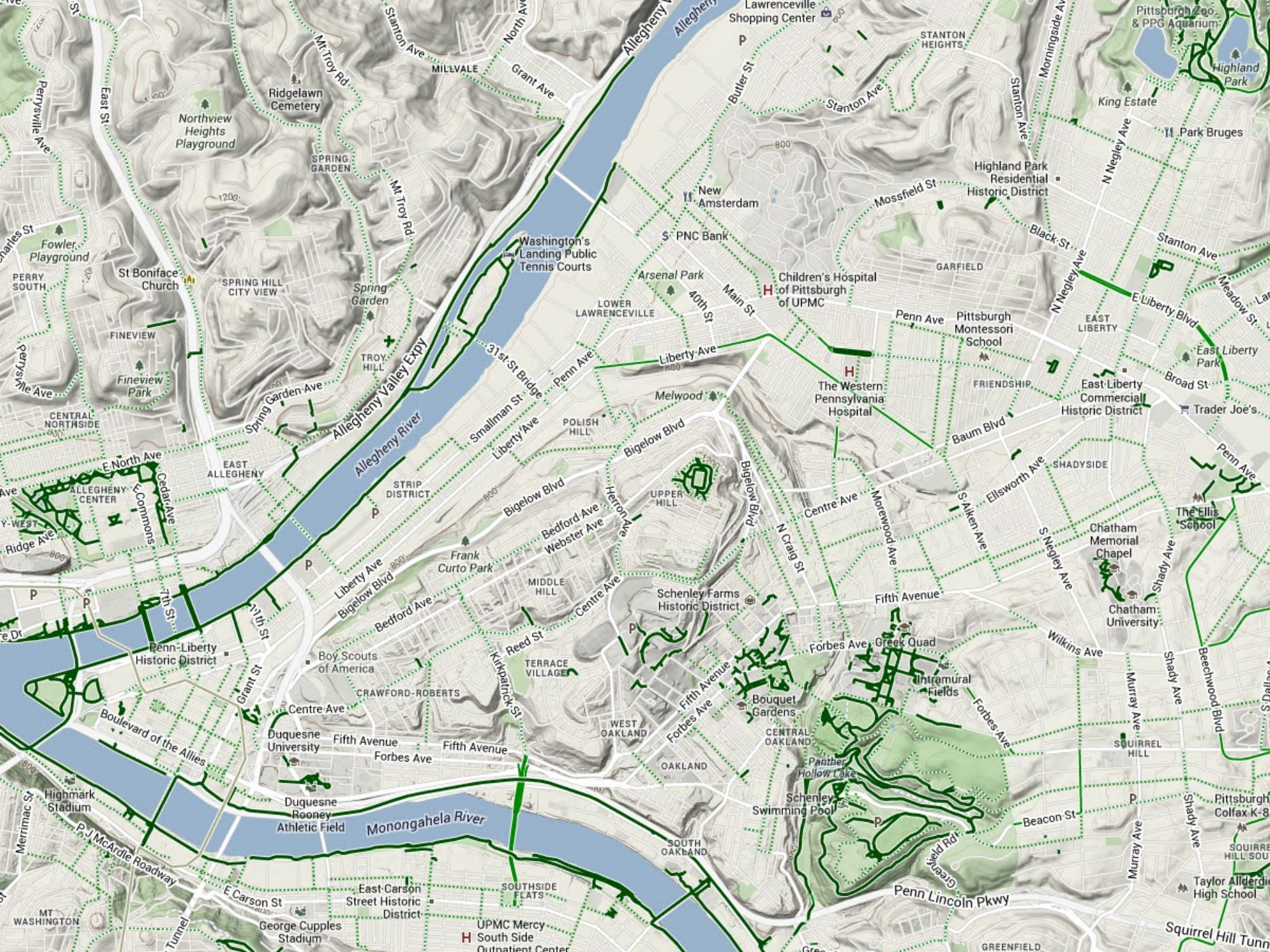


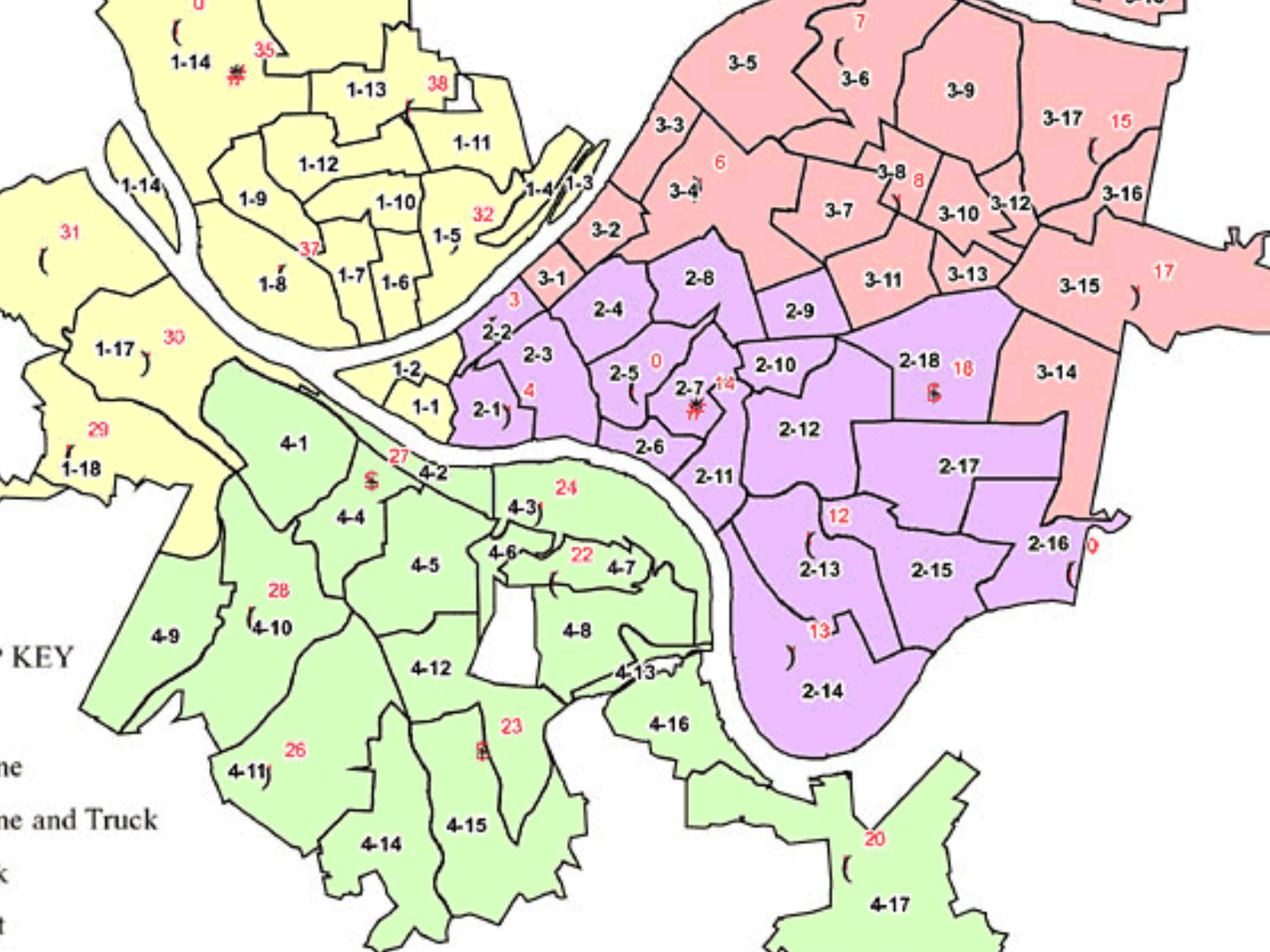
Architecture as (design) process
(activities around the other two)

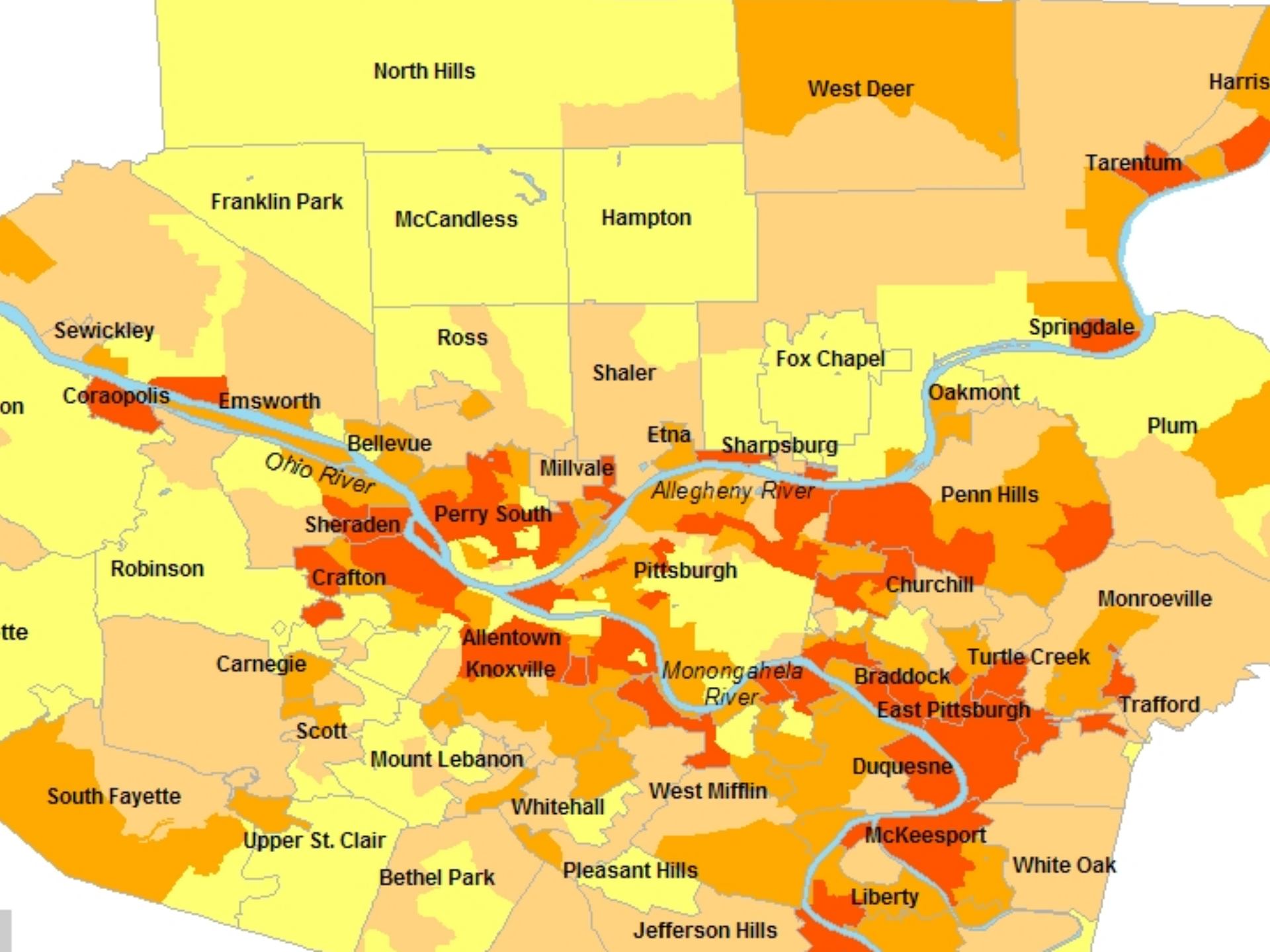
Why Document Architecture?

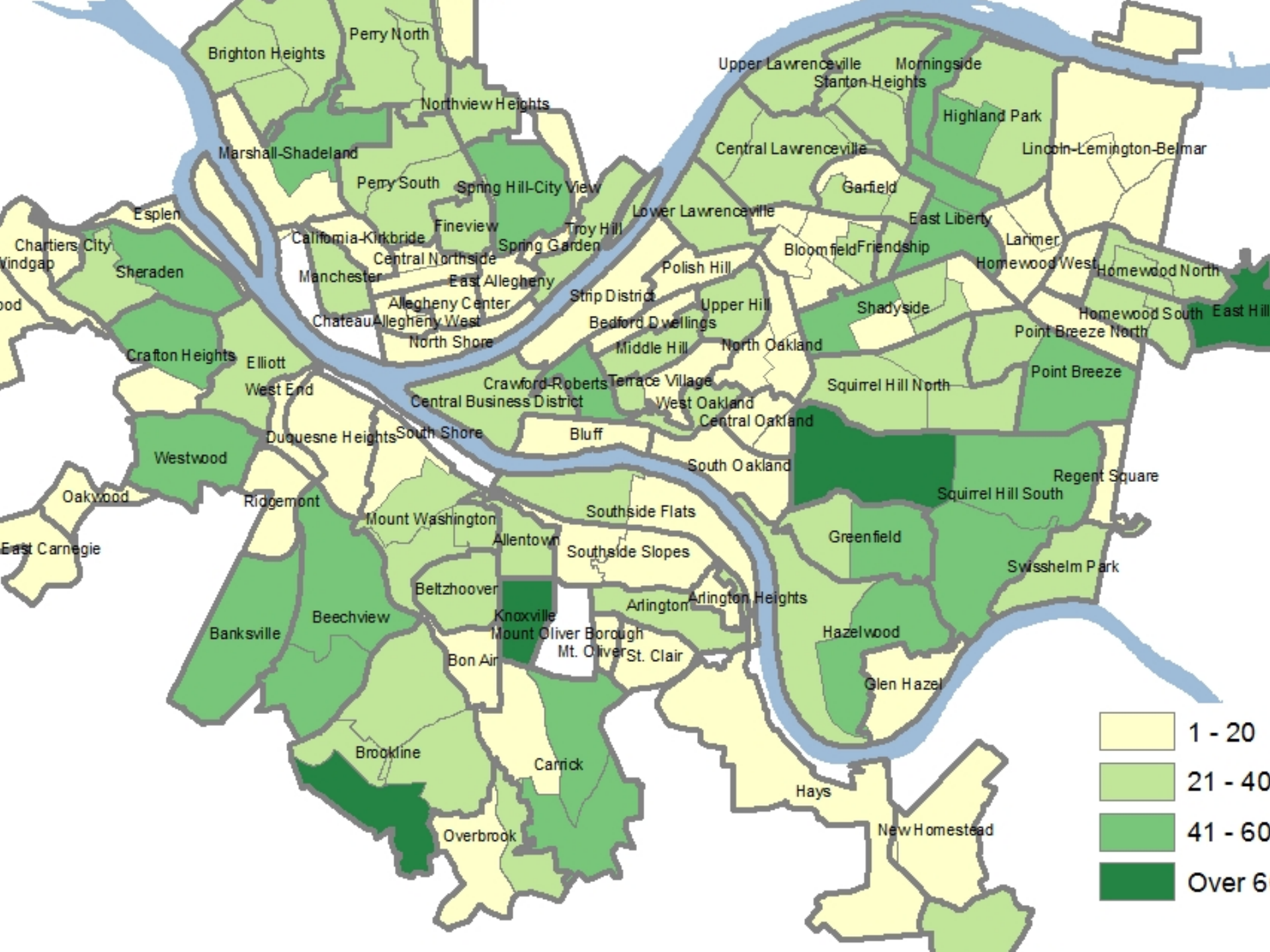
- Blueprint for the system
  - Artifact for early analysis
  - Primary carrier of quality attributes
  - Key to post-deployment maintenance and enhancement
- Documentation speaks for the architect, today and 20 years from today
  - As long as the system is built, maintained, and evolved according to its documented architecture
- Support traceability
- **Reason about quality *tradeoffs*.**


Lawrenceville Shopping Center
Pittsburgh Zoo & PPG Aquarium
Highland Park
King Estate
Park Bruges
Ridgelawn Cemetery
Northview Heights Playground
Fowler Playground
St Boniface Church
Spring Garden
Herrs Island
Washington's Landing Public Tennis Courts
New Amsterdam
PNC Bank
Arsenal Park
Children's Hospital of Pittsburgh of UPMC
Highland Park Residential Historic District
Pittsburgh Montessori School
East Liberty Park
East Liberty Commercial Historic District
Trader Joe's
The Western Pennsylvania Hospital
Melwood
Shadyside Health Education & Research Corp
The Inn on Negley
The Ellis School
Chatham Memorial Chapel
Chatham University
Frick Art & Historical Center
Fineview Park
Frank Curto Park
Schenley Farms Historic District
Penn-Liberty Historic District
Boy Scouts of America
Duquesne University
Mercy Hospital
Duquesne Rooney Athletic Field
Highmark Stadium
Greek Quad
Intramural Fields
Bouquet Gardens
Panther Hollow Lake
Schenley Swimming Pool
The Homewood Cemetery
Pittsburgh Colfax K-8
Taylor Allderdice High School
George Cupples Stadium
East Carson Street Historic District
UPMC Mercy South Side Outpatient Center
UPMC Sports
Giant Eagle - Greenfield
Pittsburgh Minadeo Elementary School
Crescent Park
Nine Mile Run
Allegheny River
Monongahela River
Allegheny Valley Expy
Penn Lincoln Pkwy
Squirrel Hill Tunnel
Liberty Tunnel
P J McArdle Roadway
Boulevard of the Allies
31st St Bridge
Washington Blvd
Fifth Avenue
Forbes Ave
Penn Ave
Liberty Ave
Bigelow Blvd
Centre Ave
Baum Blvd
Penn Ave
Fifth Avenue
Forbes Ave
Beacon St
Greenfield Ave
Josephine St
E Carson St
Millvale
Spring Garden
Troy Hill
Fineview
Central Northside
East Allegheny
Allegheny Center
Strip District
Polish Hill
Upper Hill
Middle Hill
Terrace Village
Crawford-Roberts
West Oakland
Oakland
Central Oakland
South Oakland
Southside Flats
Allentown
Mt Washington
Stanton Heights
Garfield
East Liberty
Friendship
Shadyside
Squirrel Hill
Squirrel Hill South
Greenfield
Lower Lawrenceville
Larimer
Homewood
Lincoln-Lemington-Belmar


Lawrenceville Shopping Center
Pittsburgh Zoo & PPG Aquarium
STANTON HEIGHTS
Highland Park
King Estate
Park Bruges
Highland Park Residential Historic District
Ridgelawn Cemetery
Northview Heights Playground
MILLVALE
Stanton Ave
Grant Ave
Mt Troy Rd
East St
Perrysville Ave
SPRING GARDEN
New Amsterdam
PNC Bank
Washington's Landing Public Tennis Courts
Fowler Playground
PERRY SOUTH
St Boniface Church
SPRING HILL CITY VIEW
Spring Garden
Arsenal Park
Children's Hospital of UPMC
LOWER LAWRENCEVILLE
40th St
Main St
GARFIELD
Penn Ave
Pittsburgh Montessori School
EAST LIBERTY
Liberty Ave
East Liberty Park
FINEVIEW
TROY HILL
Fineview Park
31st St Bridge
Smallman St
Spring Garden Ave
Allegheny Valley Expy
Allegheny River
Melwood
The Western Pennsylvania Hospital
FRIENDSHIP
East Liberty Commercial Historic District
Trader Joe's
Broad St
CENTRAL NORTHSIDE
POLISH HILL
Bigelow Blvd
Baum Blvd
EAST ALLEGHENY
E North Ave
ALLEGHENY CENTER
STRIP DISTRICT
UPPER HILL
SHADYSIDE
The Ellis School
Cedar Ave
Commons
Ridge Ave
Herron Ave
Bedford Ave
Webster Ave
Centre Ave
Morewood Ave
S Aiken Ave
Ellsworth Ave
S Negley Ave
Chatham Memorial Chapel
Shady Ave
Frank Curto Park
MIDDLE HILL
Schenley Farms Historic District
Fifth Avenue
Chatham University
Penn-Liberty Historic District
Boy Scouts of America
TERRACE VILLAGE
Reed St
Kirkpatrick St
Grant St
7th St
11th St
Forbes Ave
Greek Quad
Wilkins Ave
Intramural Fields
Beechwood Blvd
CRAWFORD-ROBERTS
Bouquet Gardens
Boulevard of the Allies
Duquesne University
Fifth Avenue
WEST OAKLAND
CENTRAL OAKLAND
SQUIRREL HILL
Panther Hollow Lake
OAKLAND
Schenley Swimming Pool
Beacon St
Pittsburgh Colfax K-8
Highmark Stadium
Duquesne Rooney Athletic Field
Monongahela River
SOUTH OAKLAND
P.J. McArdle Roadway
Greenfield Rd
Penn Lincoln Pkwy
Taylor Allderdice High School
East Carson Street Historic District
SOUTHSIDE FLATS
E Carson St
MT WASHINGTON
George Cupples Stadium
UPMC Mercy South Side Outpatient Center
GREENFIELD
Squirrel Hill Tunn


1-14
35
1-13
38
1-11
1-12
1-14
1-9
1-10
1-4
1-3
32
1-5
31
37
1-8
1-7
1-6
1-17
30
1-2
1-1
29
1-18
4-1
27
4-2
4-4
24
4-3
4-5
4-6
22
4-7
28
4-10
4-9
4-12
4-8
4-13
23
4-11
26
4-15
4-14
4-16
20
4-17
3-5
7
3-6
3-9
3-17
15
3-3
6
3-4
3-8
8
3-7
3-10
3-12
3-16
3-2
3-1
2-8
3-11
3-13
3-15
17
3
2-4
2-9
2-2
2-3
2-5
0
2-7
14
2-10
2-18
16
3-14
4
2-1
2-12
2-6
2-11
2-17
12
2-16
0
2-13
2-15
13
2-14
KEY
ne
ne and Truck


North Hills
West Deer
Harris
Tarentum
Franklin Park
McCandless
Hampton
Springdale
Sewickley
Ross
Shaler
Fox Chapel
Oakmont
Coraopolis
Emsworth
Plum
Bellevue
Etna
Sharpsburg
Ohio River
Millvale
Allegheny River
Penn Hills
Perry South
Sheraden
Robinson
Crafton
Pittsburgh
Churchill
Monroeville
Allentown
Knoxville
Carnegie
Monongahela River
Braddock
Turtle Creek
East Pittsburgh
Trafford
Scott
Mount Lebanon
Duquesne
South Fayette
West Mifflin
Whitehall
McKeesport
Upper St. Clair
White Oak
Bethel Park
Pleasant Hills
Liberty
Jefferson Hills


Perry North
Brighton Heights
Northview Heights
Marshall-Shadeland
Perry South
Spring Hill-City View
Esplen
Fineview
Troy Hill
California-Kirkbride
Spring Garden
Chartiers City
Central Northside
Sheraden
Manchester
East Allegheny
Allegheny Center
Chateau
Allegheny West
North Shore
Crafton Heights
Elliott
West End
Duquesne Heights
South Shore
Westwood
Oakwood
Ridgemont
East Carnegie
Mount Washington
Allentown
Beltzhoover
Knoxville
Beechview
Banksville
Bon Air
Mount Oliver Borough
Mt. Oliver
Brookline
Carrick
Overbrook
Crawford-Roberts
Central Business District
Strip District
Polish Hill
Lower Lawrenceville
Upper Lawrenceville
Morningside
Stanton Heights
Highland Park
Central Lawrenceville
Lincoln-Lemington-Belmar
Garfield
East Liberty
Larimer
Bloomfield
Friendship
Homewood West
Homewood North
Homewood South
East Hill
Shadyside
Point Breeze North
Point Breeze
Upper Hill
Bedford Dwellings
Middle Hill
North Oakland
Terrace Village
West Oakland
Central Oakland
Squirrel Hill North
Squirrel Hill South
Regent Square
Bluff
South Oakland
Southside Flats
Southside Slopes
Arlington
Arlington Heights
Greenfield
Swisshelm Park
Hazelwood
Glen Hazel
St. Clair
Hays
New Homestead
1 - 20
21 - 40
41 - 60
Over 6


Pittsburgh
Allegheny Cemetery
Highland Park
Pittsburgh Zoo
Allegheny River
Butler St
Stanton Ave
Chislett St
N Negley Ave
N Euclid Ave
Lake Dr
Washington Blvd
Penn Ave
Liberty Ave
31st St
Smallman St
Spring Garden Ave
Pitview Ave
Troy Rd
East St
E Ohio St
River Ave
Spring Way
Bigelow Blvd
Webster Ave
Centre Ave
Baum Blvd
E Busway
Ellsworth Ave
Larimer Ave
Lincoln Ave
Brushton Ave
Hamilton Ave
Bennett St
Mellon Park
5th Ave
Wilkins Ave
Beechwood Blvd
Homewood Cemetery
Wilkinsburg
Penn Ave
West Park
Heinz Field
Clemente Park
Point State Park
Liberty Ave
Carson St
Duquesne University
Forbes Ave
2nd Ave
University of Pittsburgh
Carnegie Mellon University
Schenley Park
Frick Park
S Braddock Ave
Ninemile Run
Murray Ave
Hazelwood Ave
Calvary Cemetery
Johnston Ave
E Carson St
Josephine St
Southside Park
Arlington Ave
Liberty Tunnels
Saw Mill Run Blvd
McKinley Park
Moore Park
Becks Run Rd
E Agnew St
Brookline Blvd
Edgebrook Ave
Pat Busway
Glass Run
E Waterfront Dr
The Waterfront
E 8th Ave
Amity St
Ann St
Munhall
Greensprings
Swissv
279
28
579
376
22
885
51
837
8

Describing a system's architecture typically requires multiple views or viewpoints.

- Multiple views reflect different perspectives on the design problem.
- Every view should align with a purpose
- Different views are suitable for different reasoning aspects (different quality goals), e.g.,
  - Performance
  - Extensibility
  - Security
  - Scalability
  - …

Two views of a client-server system

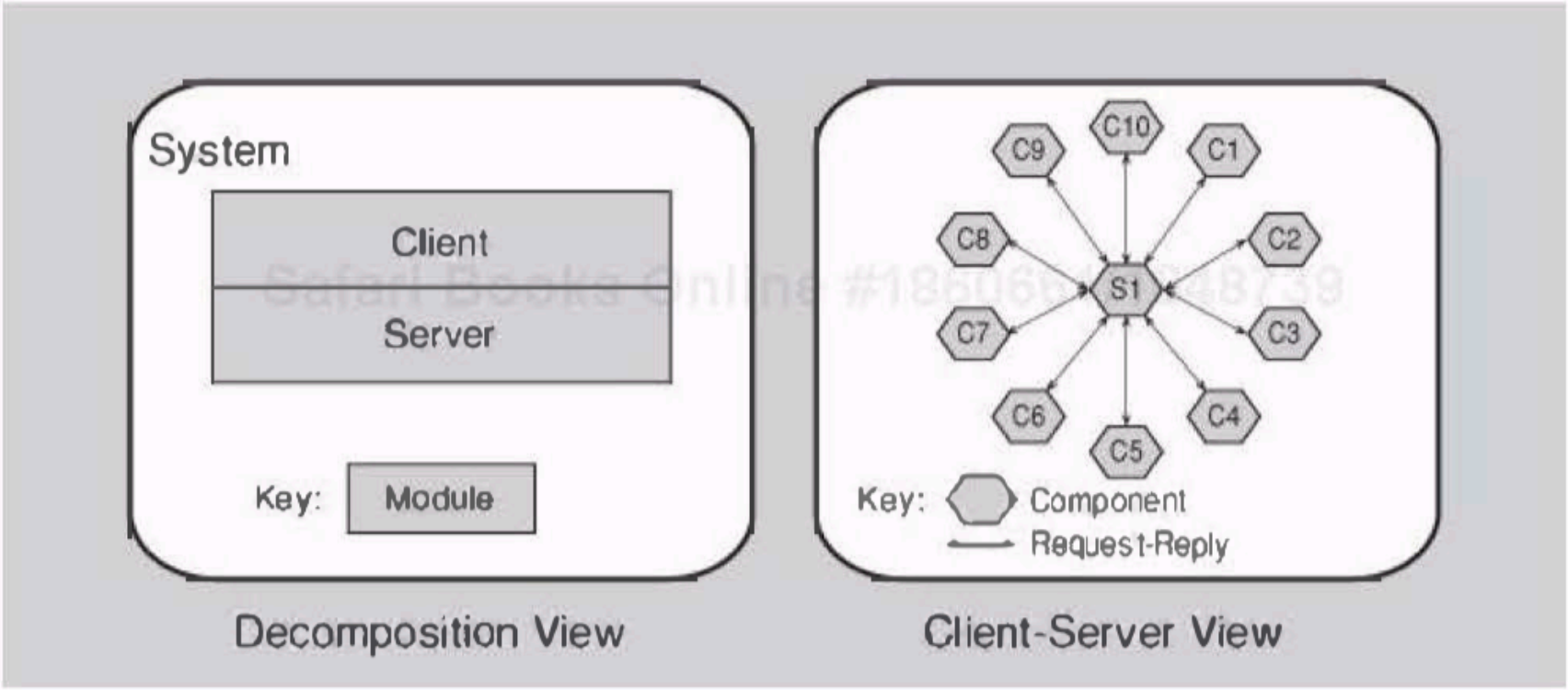
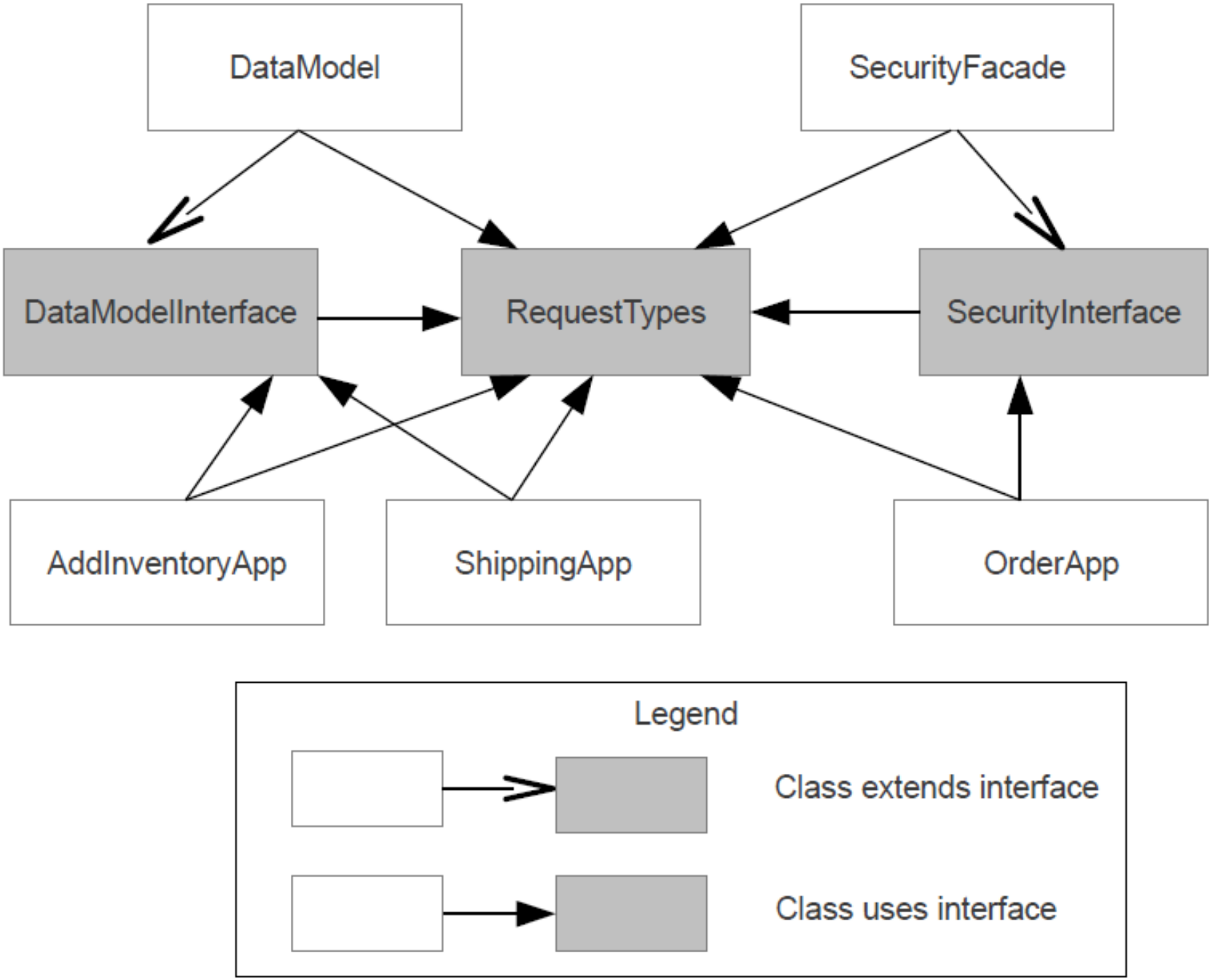
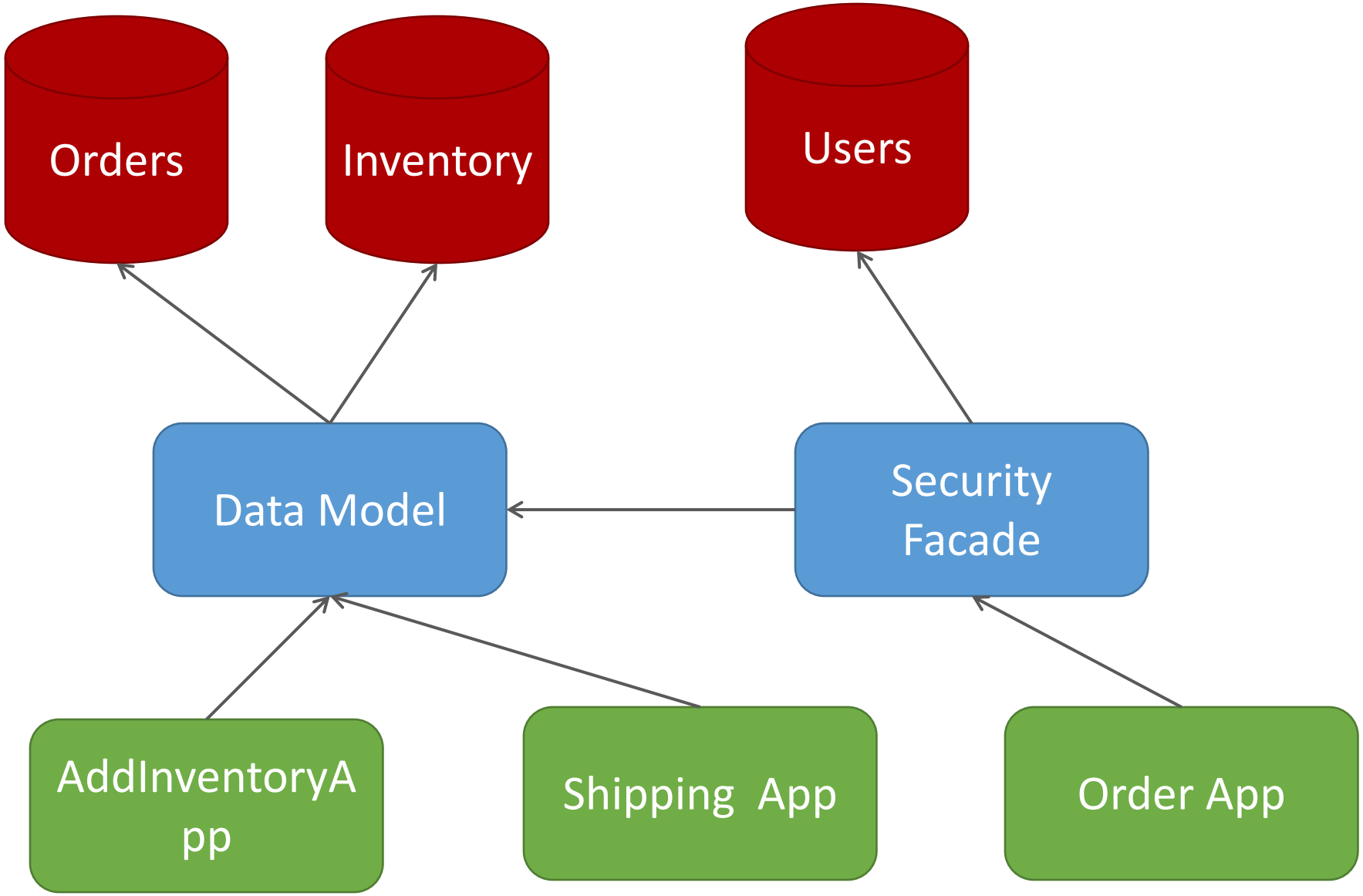


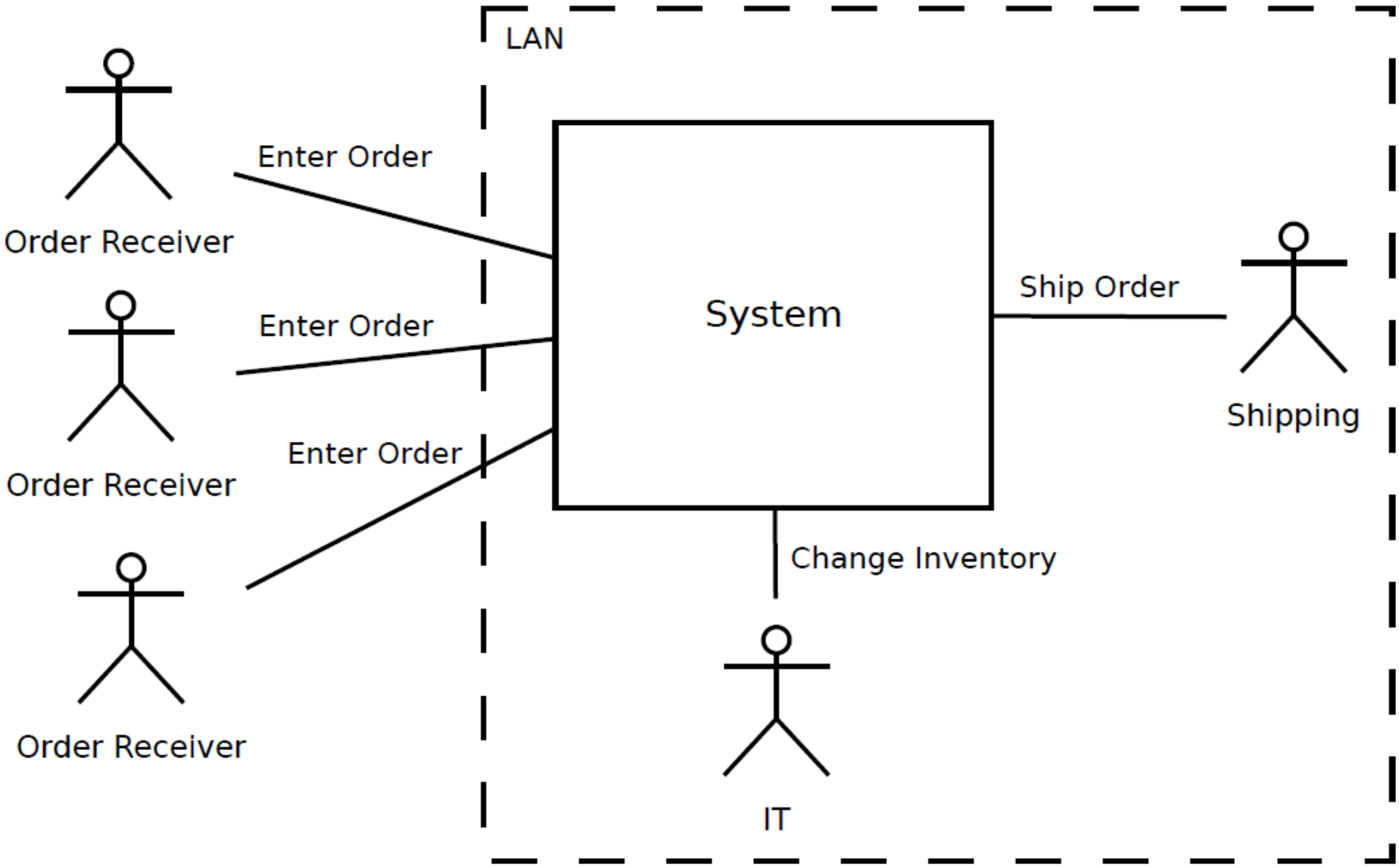
Figure 1.2. Two views of a client-server system

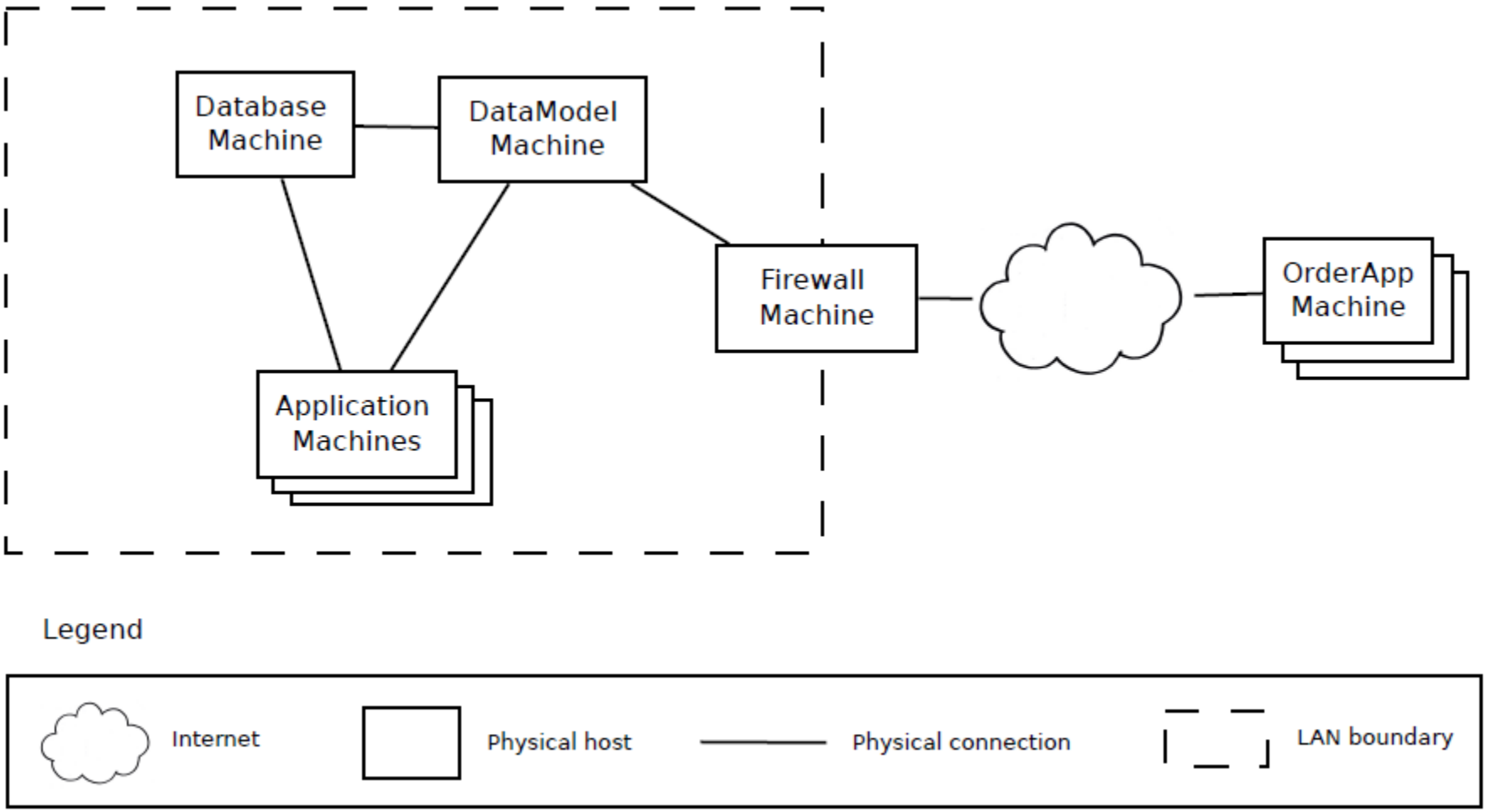
Common Views in Documenting Software Architecture

- Static View
  - Modules (subsystems, structures) and their relations (dependencies, …)
- Dynamic View
  - Components (processes, runnable entities) and connectors (messages, data flow, …)
- Physical View (Deployment)
  - Hardware structures and their connections


DataModel
SecurityFacade
DataModelInterface
RequestTypes
SecurityInterface
AddInventoryApp
ShippingApp
OrderApp
Legend
Class extends interface
Class uses interface


Orders
Inventory
Users
Data Model
Security Facade
AddInventoryApp
Shipping App
Order App


LAN
Enter Order
Order Receiver
Enter Order
Order Receiver
Enter Order
Order Receiver
System
Ship Order
Shipping
Change Inventory
IT


Database Machine
DataModel Machine
Firewall Machine
OrderApp Machine
Application Machines
Legend
Internet
Physical host
Physical connection
LAN boundary

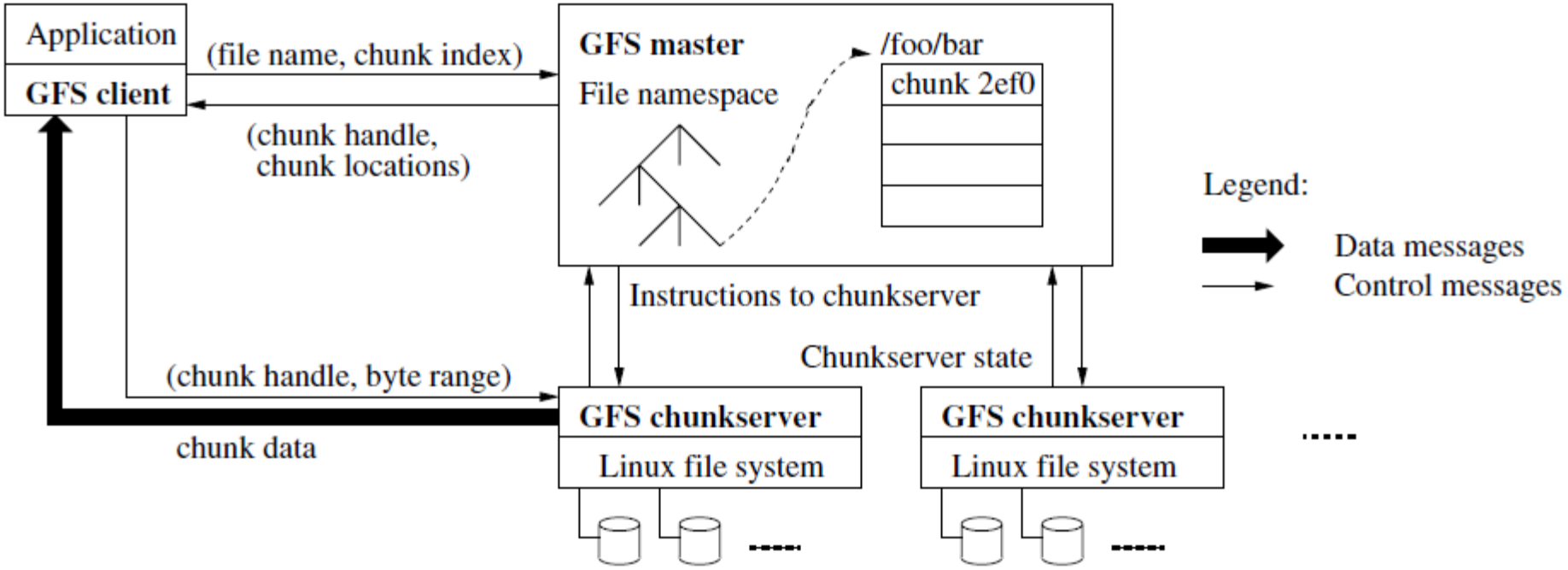


Figure 1: GFS Architecture

Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." *ACM SIGOPS operating systems review*. Vol. 37. No. 5. ACM, 2003.

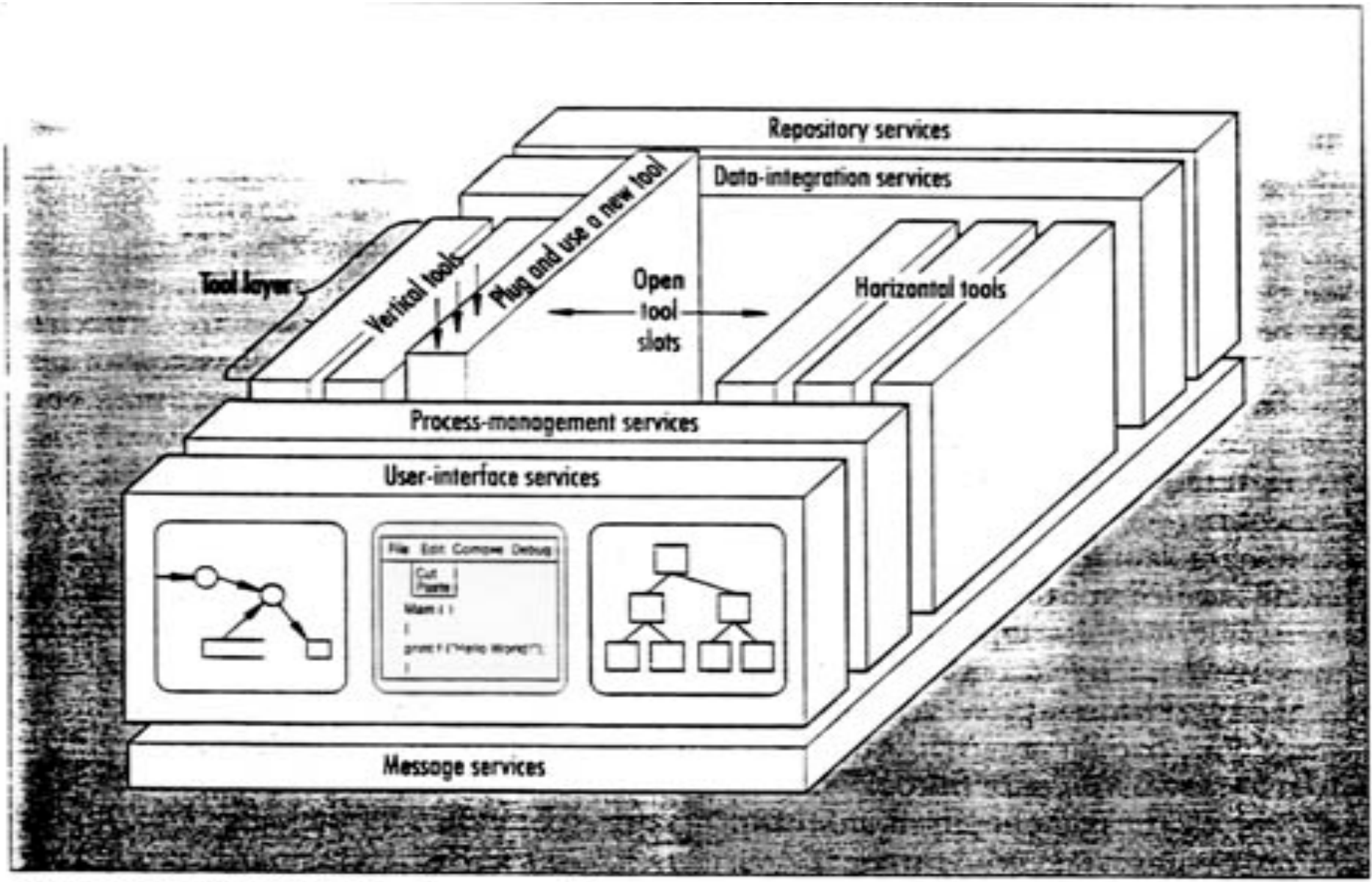


Figure 1. The NIST/ECMA reference model.

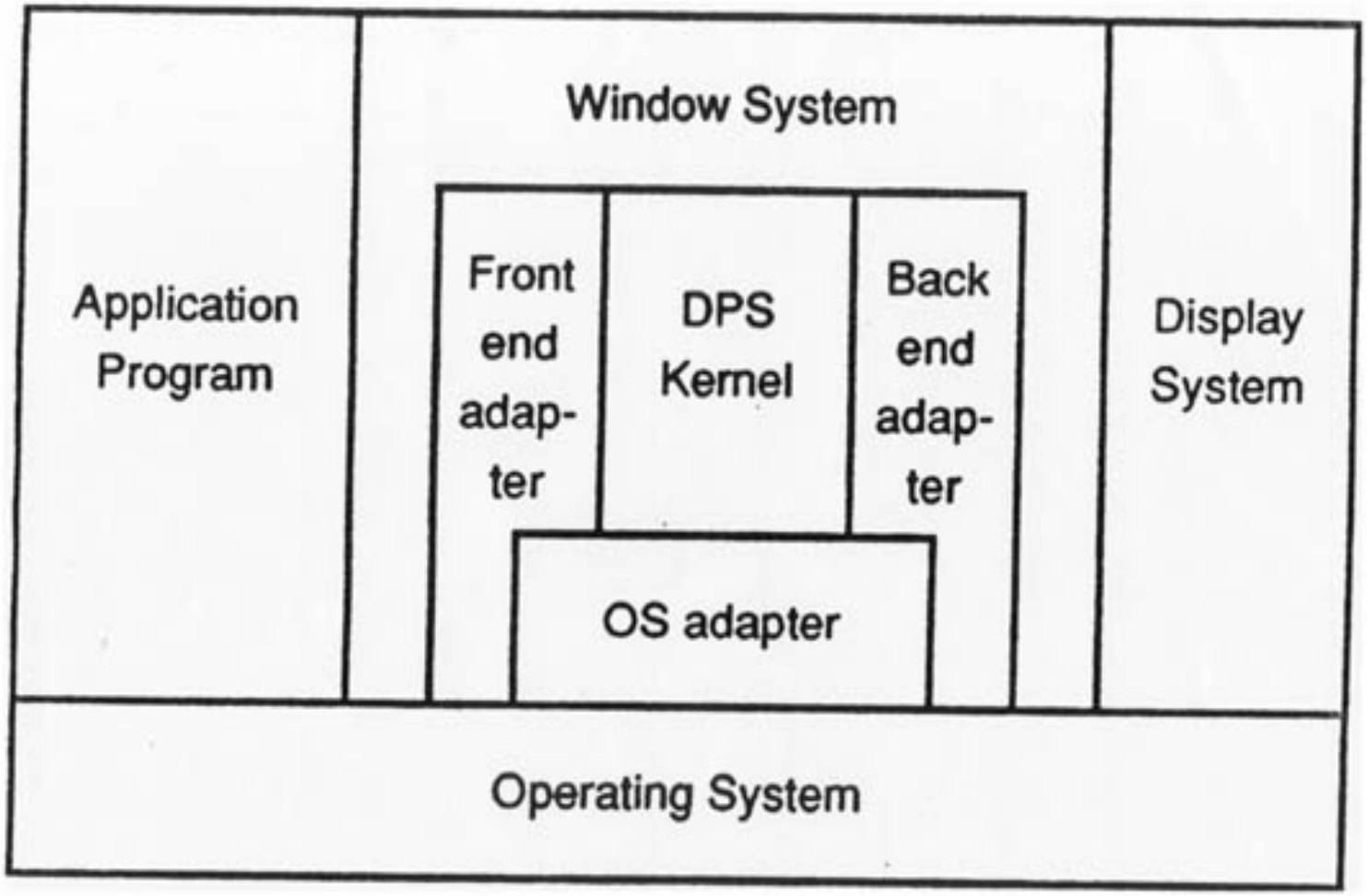


Figure 2. Display PostScript interpreter components.

An Overview of the DISPLAY POSTSCRIPT™ System, Adobe Systems Incorporated, March 16, 1988, P. 10

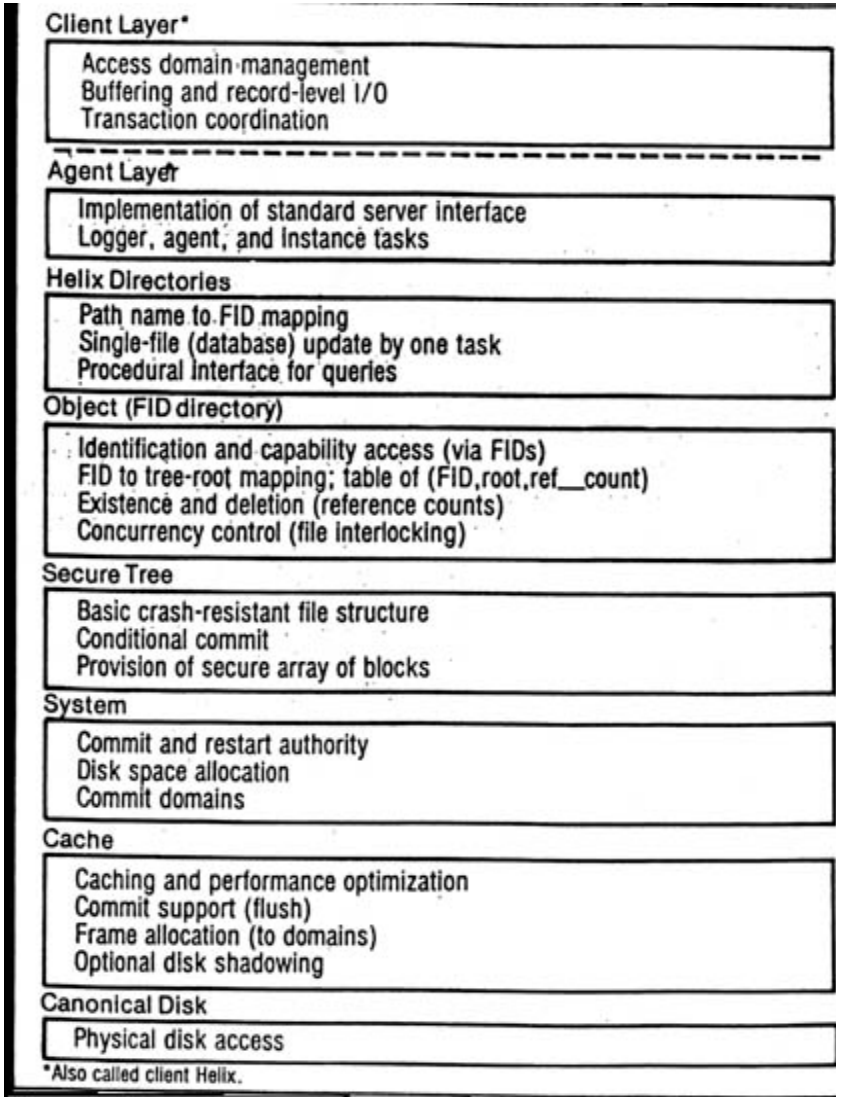


Figure 2. Abstraction layering.

IEEE Software, "Helix: The architecture of the XMS Distributed File System, Marek Fridrich and William Older, May 1985, Vol. 2, No. 3, P. 23

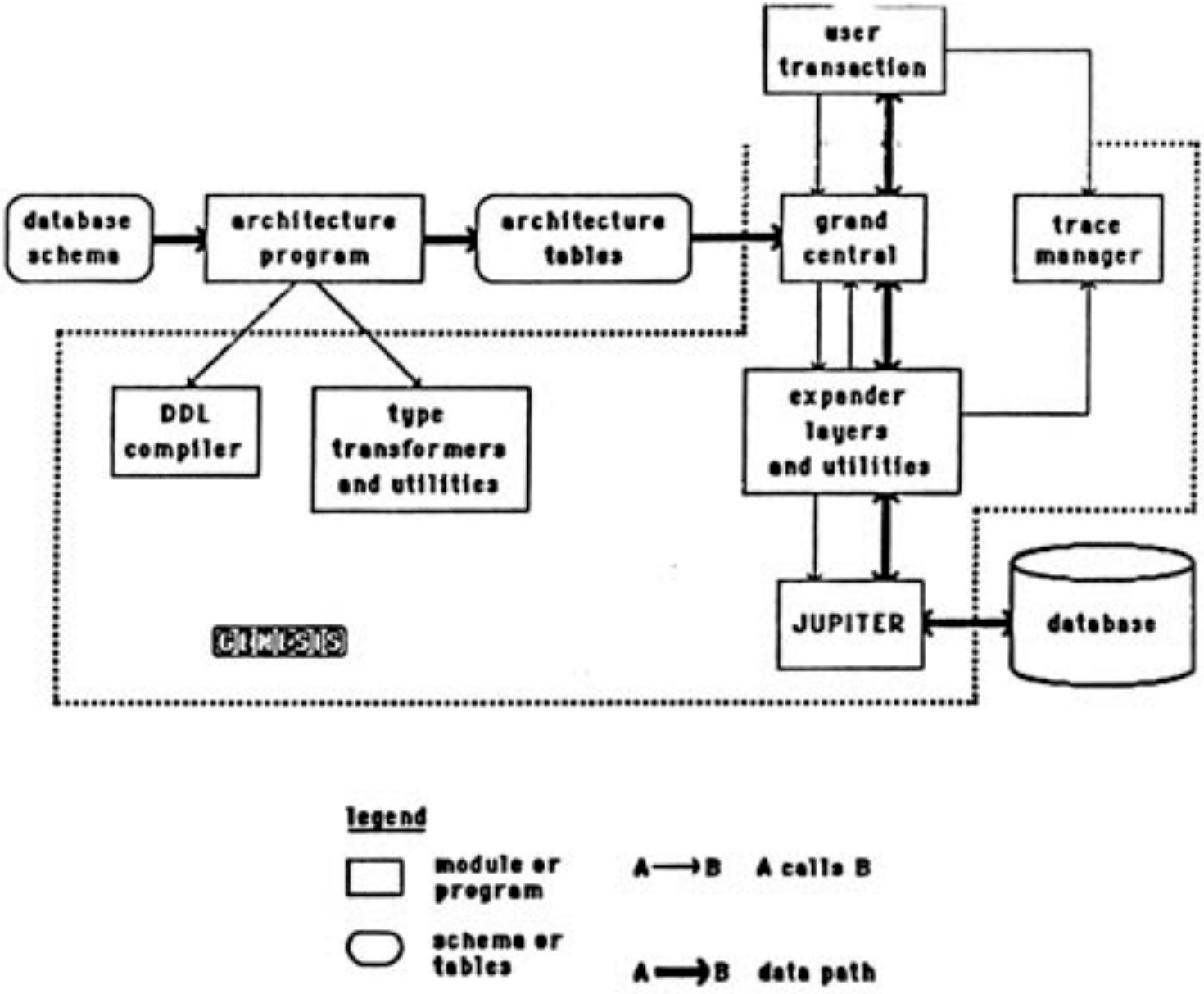
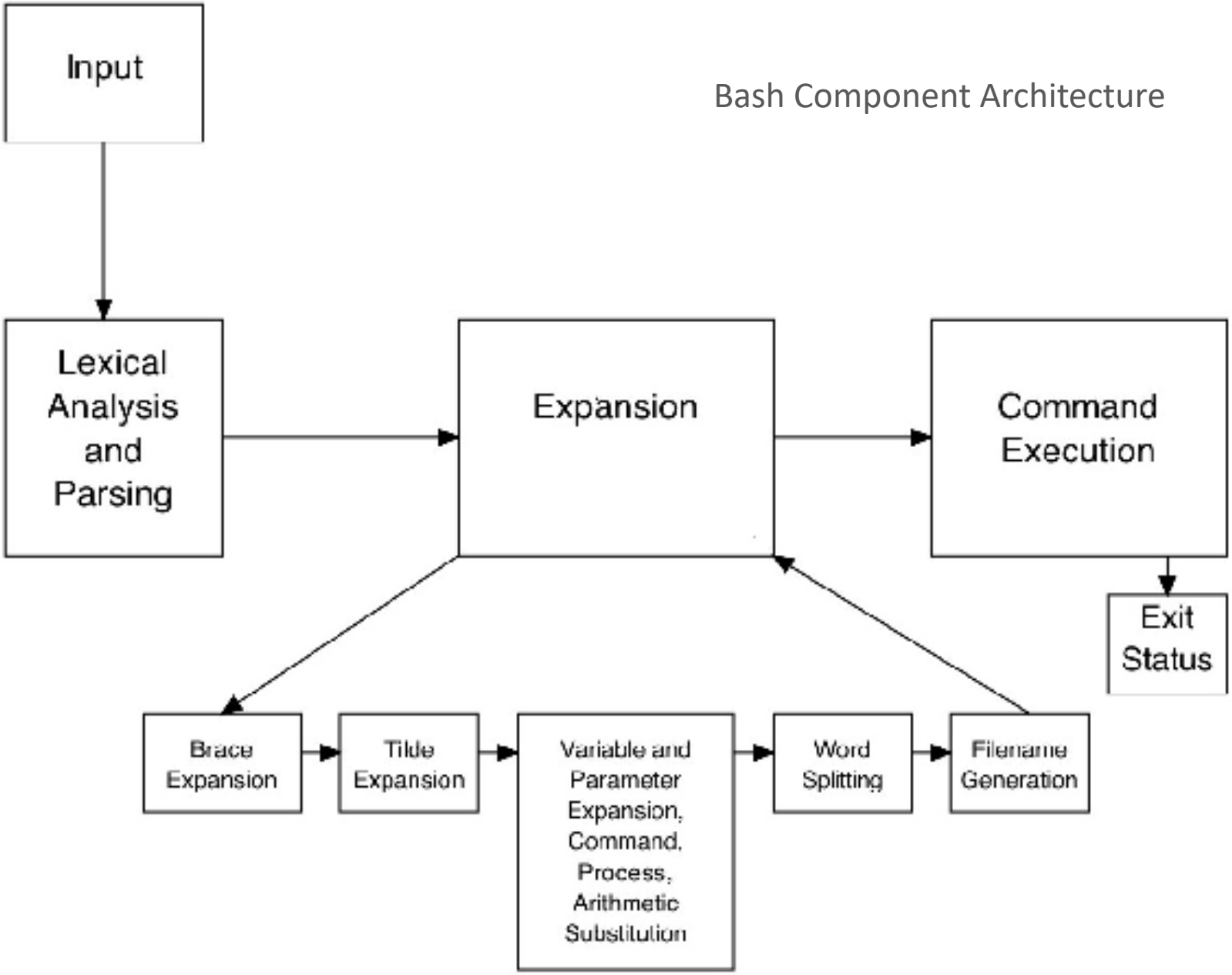


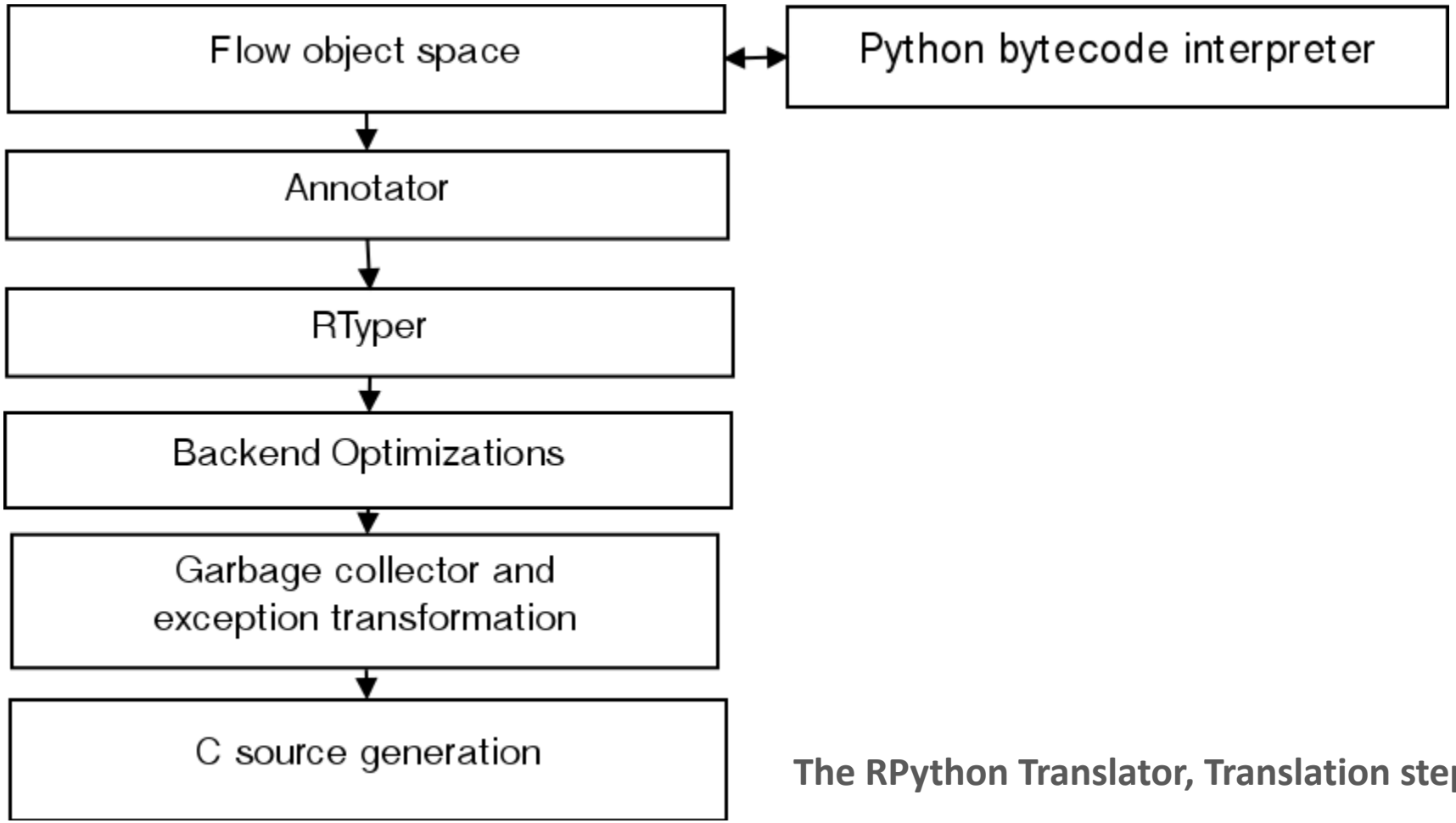
Figure 3.1 The Configuration of the GENESIS Prototype

Genesis: A Reconfiguration Database Management System, D. S. Batory, J.R. Barnett, J.F. Garza, K.P. Smith, K. Tsukuda, B.C. Twichell, T.E. Wise, Department of Computer Sciences, University of Texas at Austin,

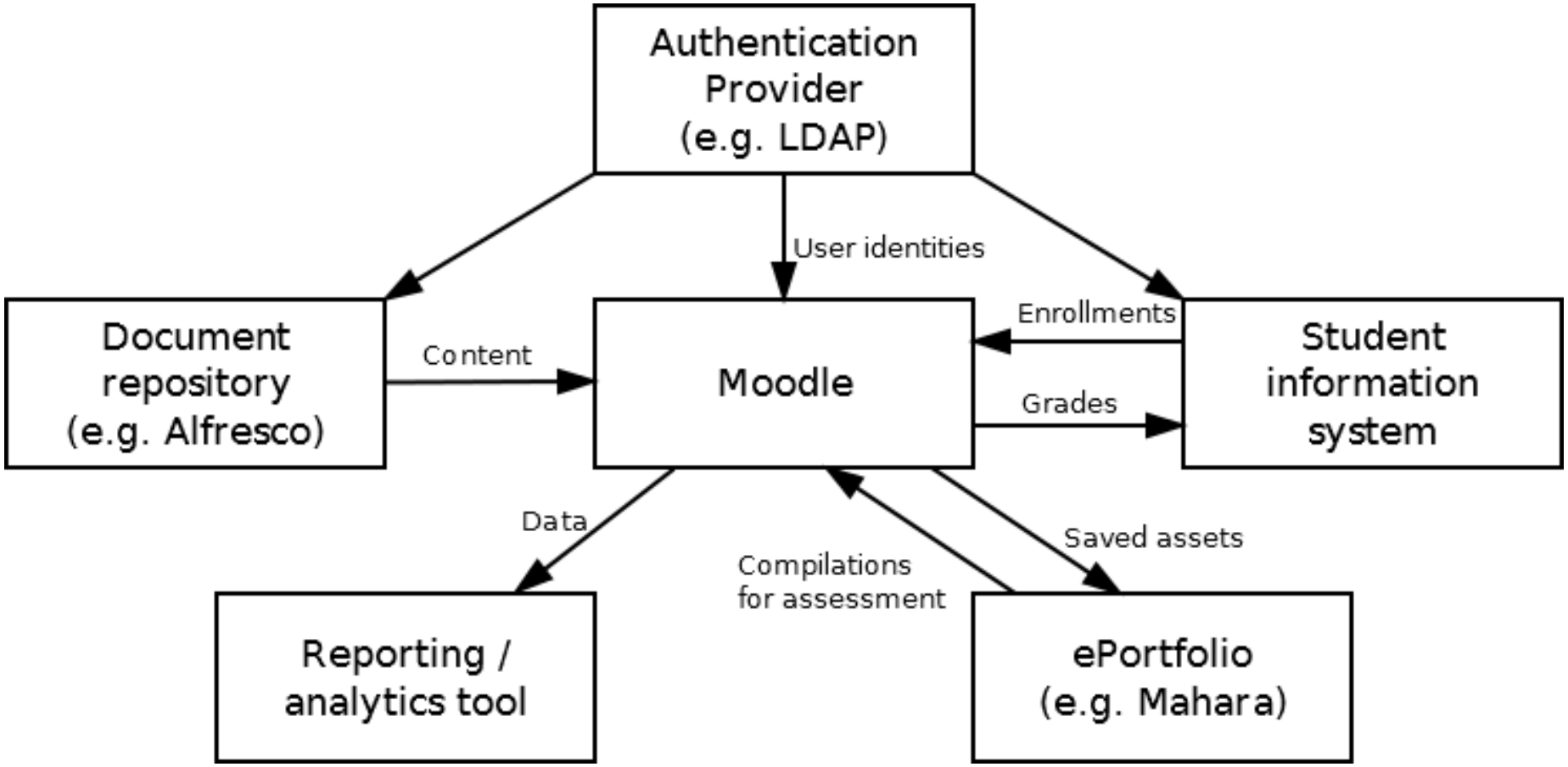
Bash Component Architecture



Example source:
http://www.aosabook.org



The RPython Translator, Translation steps



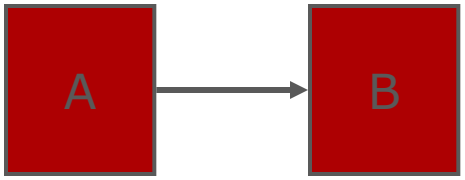
Moodle: Typical university systems architecture – Key subsystems

Bad architectural documentation.

- In practice today's documentation consists of
  - Ambiguous box-and-line diagrams
  - Inconsistent use of notations
  - Confusing combinations of viewtypes
- Many things are left unspecified:
  - What kind of elements?
  - What kind of relations?
  - What do the boxes and arrows mean?
  - What is the significance of the layout?

What could the arrow mean?

- Many possibilities
  - A passes control to B
  - A passes data to B
  - A gets a value from B
  - A streams data to B
  - A sends a message to B
  - A creates B
  - A occurs before B
  - B gets its electricity from A
  - …



Guidelines: Avoiding Ambiguity

- Always include a legend
- Define precisely what the boxes mean
- Define precisely what the lines mean
- Don't mix viewtypes unintentionally
  - Recall: Module (classes), C&C (components)
- Supplement graphics with explanation
  - Very important: rationale (architectural intent)
- Do not try to do too much in one diagram
  - Each view of architecture should fit on a page
  - Use hierarchy

CASE STUDY: THE GOOGLE FILE SYSTEM

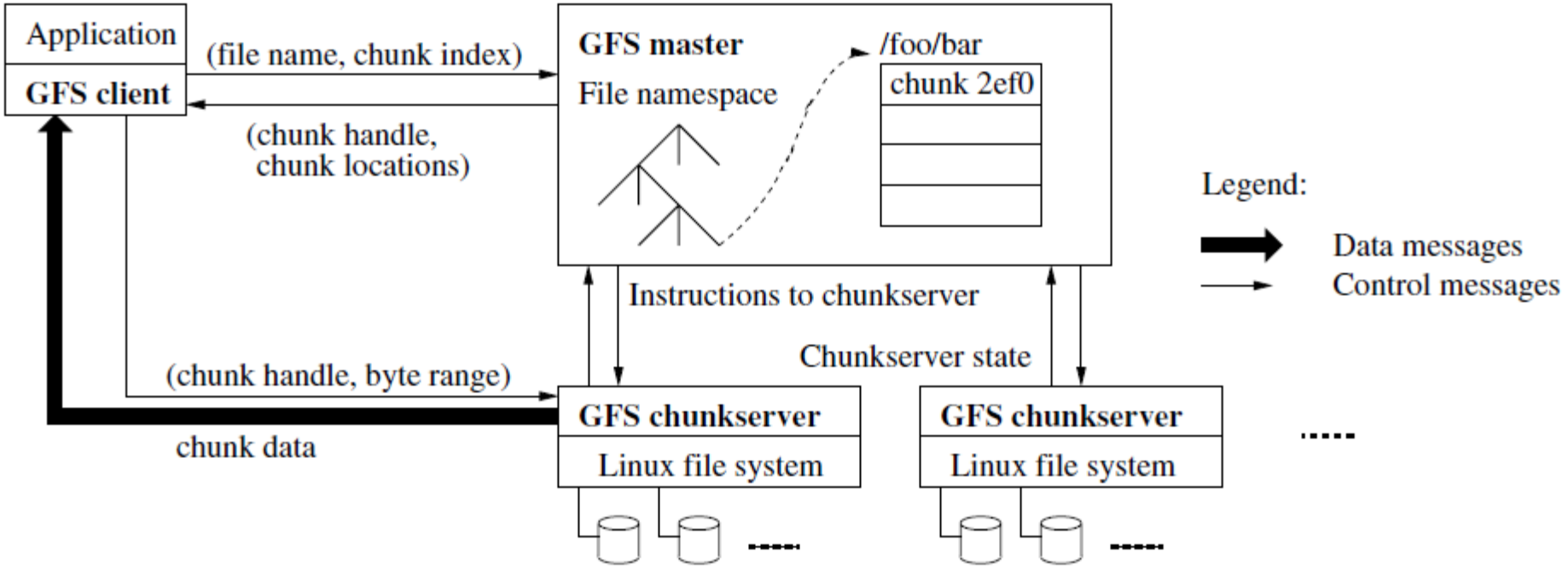


Figure 1: GFS Architecture

Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." *ACM SIGOPS operating systems review*. Vol. 37. No. 5. ACM, 2003.

Assumptions

- The system is built from many inexpensive commodity components that often fail.
- The system stores a modest number of large files.
- The workloads primarily consist of two kinds of reads: large streaming reads and small random reads.
- The workloads also have many large, sequential writes that append data to files.
- The system must efficiently implement well-defined semantics for multiple clients that concurrently append to the same file.
- High sustained bandwidth is more important than low latency.

Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." *ACM SIGOPS operating systems review*. Vol. 37. No. 5. ACM, 2003.

Qualities:
Scalability
Reliability
Performance
Cost

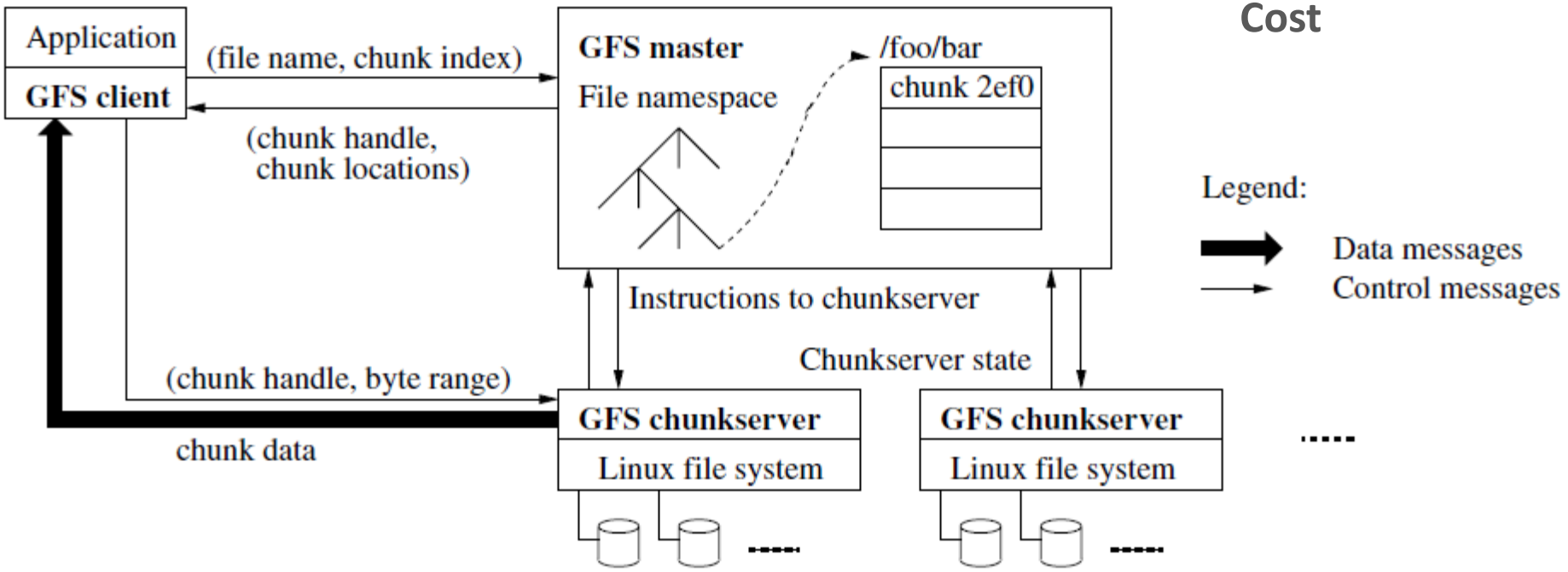


Figure 1: GFS Architecture

Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." *ACM SIGOPS operating systems review*. Vol. 37. No. 5. ACM, 2003.

Questions

1. What are the most important quality attributes in the design?
2. How are those quality attributes realized in the design?

Qualities:
Scalability
Reliability
Performance
Cost

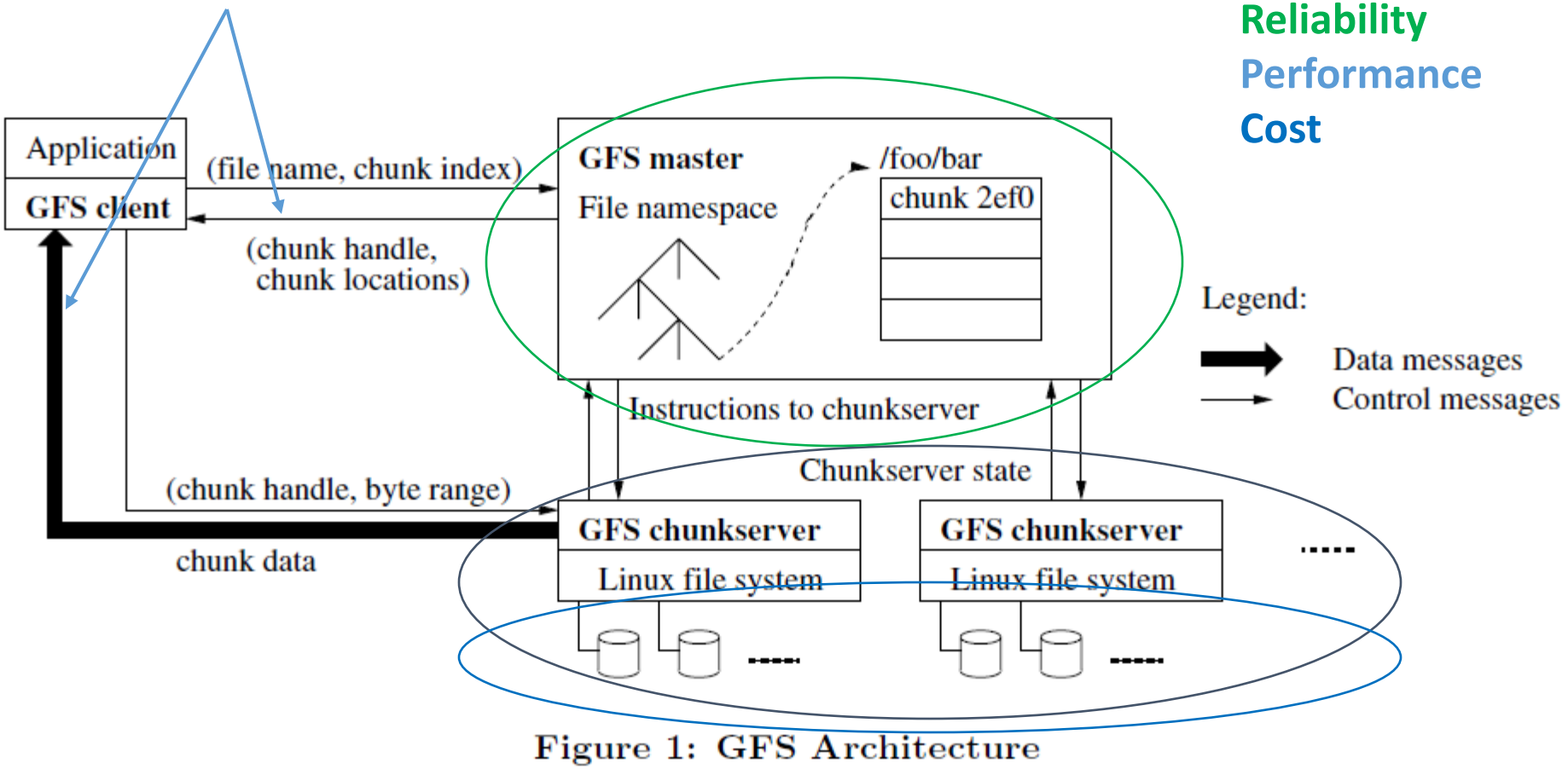


Figure 1: GFS Architecture

Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." *ACM SIGOPS operating systems review*. Vol. 37. No. 5. ACM, 2003.

Exercise

For the Google File System, create a physical architecture view that addresses a relevant quality attribute

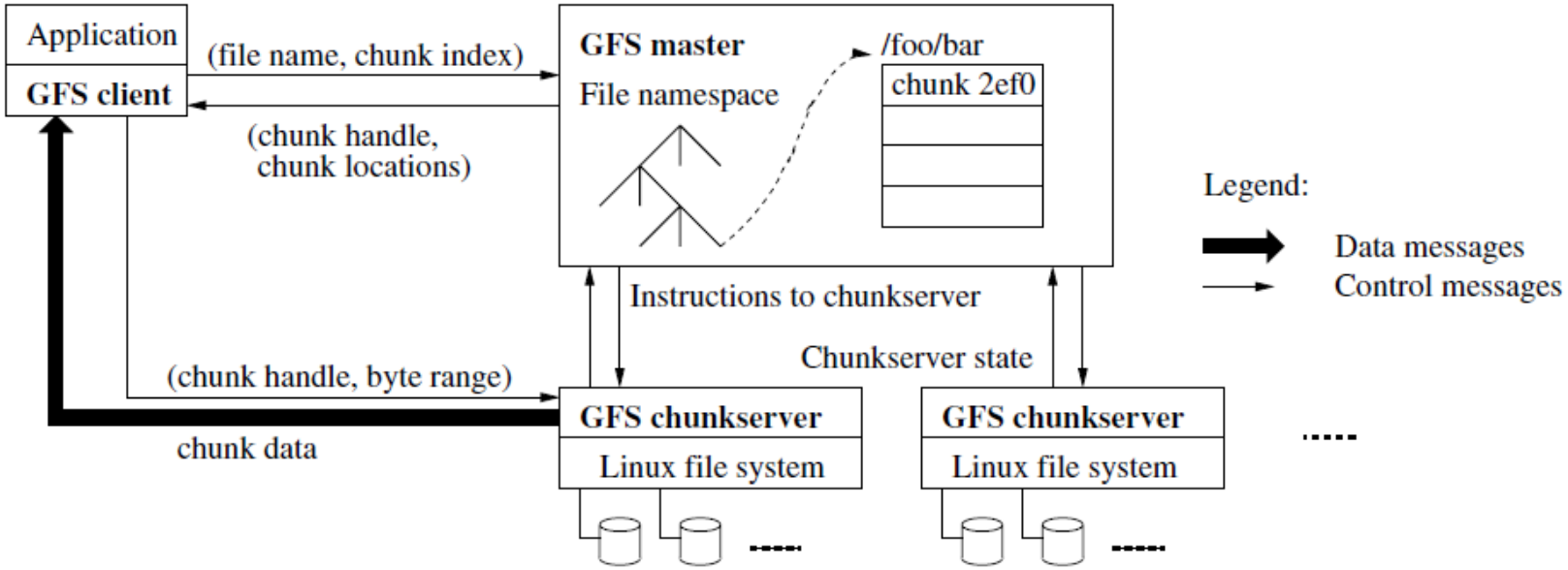
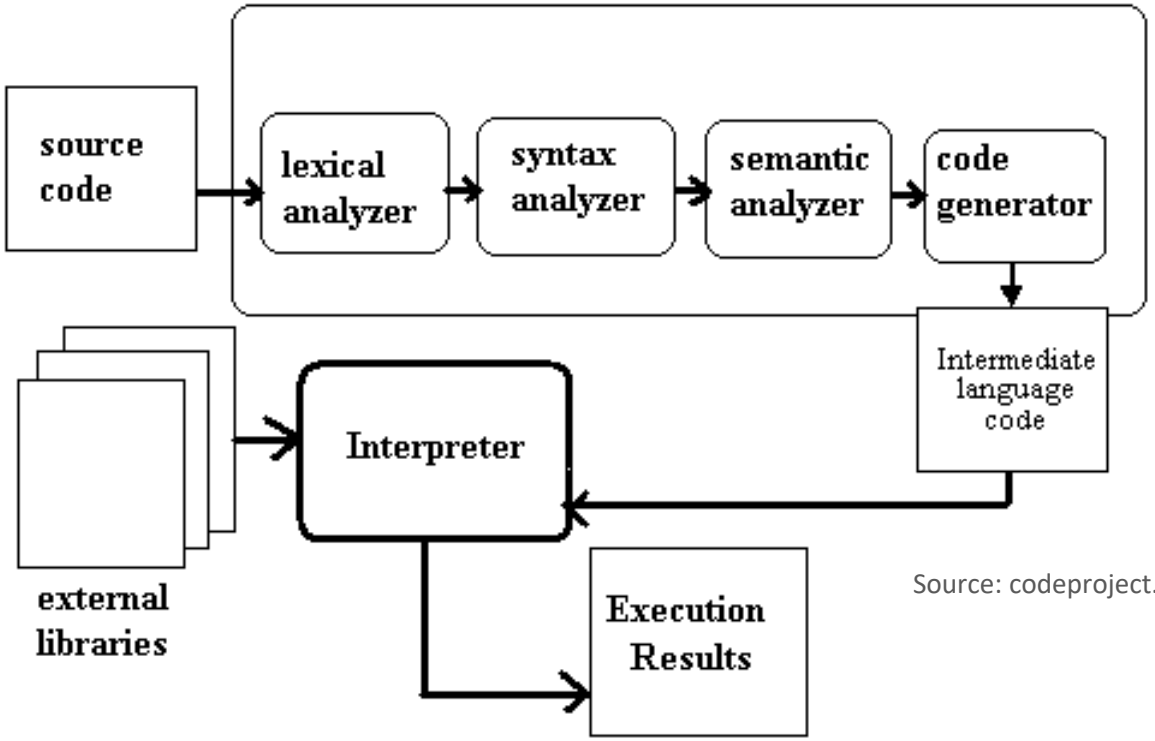


Figure 1: GFS Architecture

ARCHITECTURAL STYLES AND TACTICS

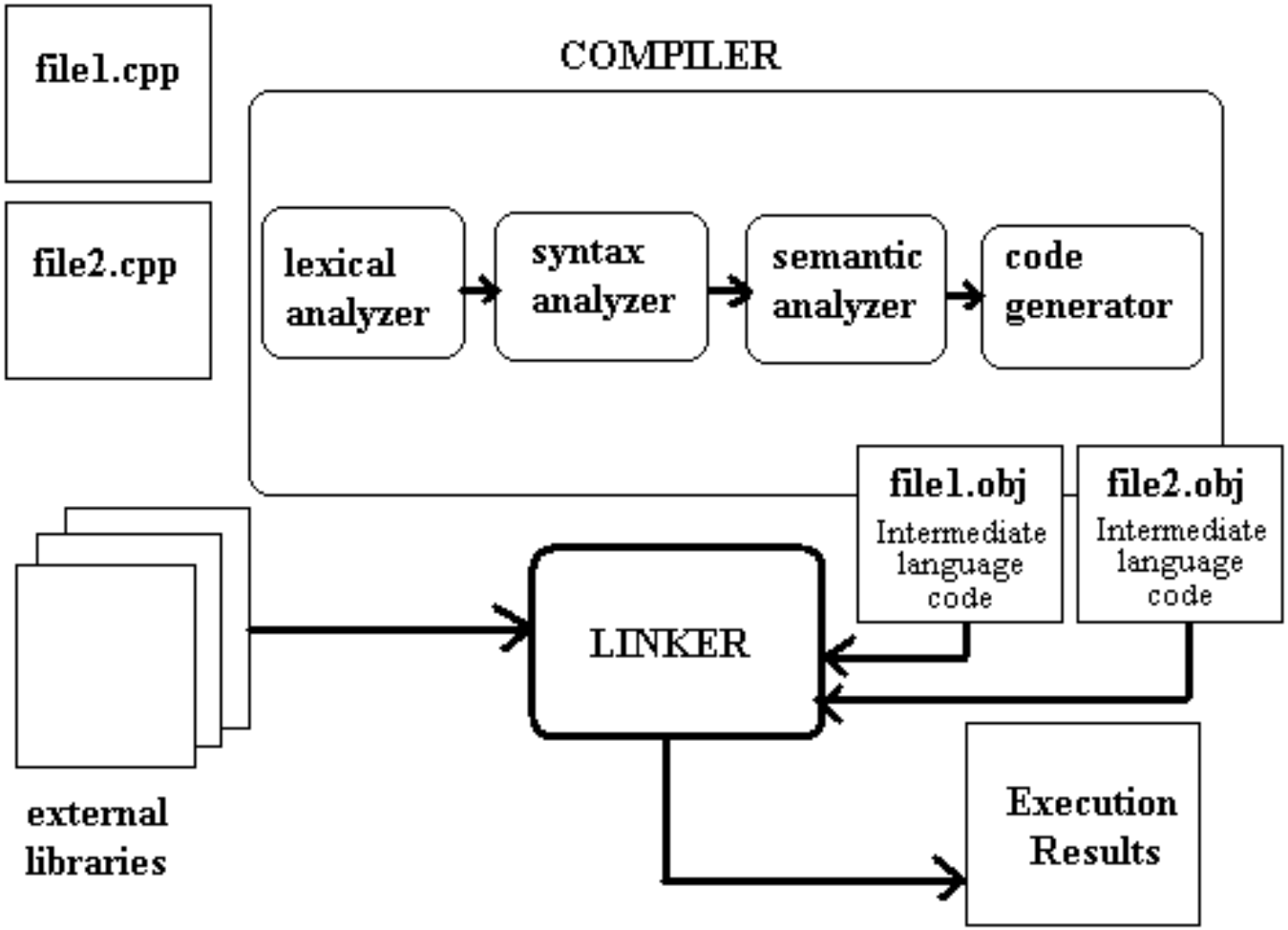
Architectural style/pattern: broad principle of system organization.

- Describes computational model
  - E.g., pipe and filter, call-return, publish-subscribe, layered, services
- Related to one of common view types
  - Static, dynamic, physical



Source: codeproject.org

Architectural style (pattern)



Source: codeproject.org

Example Architectural Patterns

- System organization
  - Repository model
  - Client-server model
  - Layered model
- Modular decomposition
  - Object oriented
  - Function-oriented pipelining
- Control styles
  - Centralized control
  - Event-driven systems

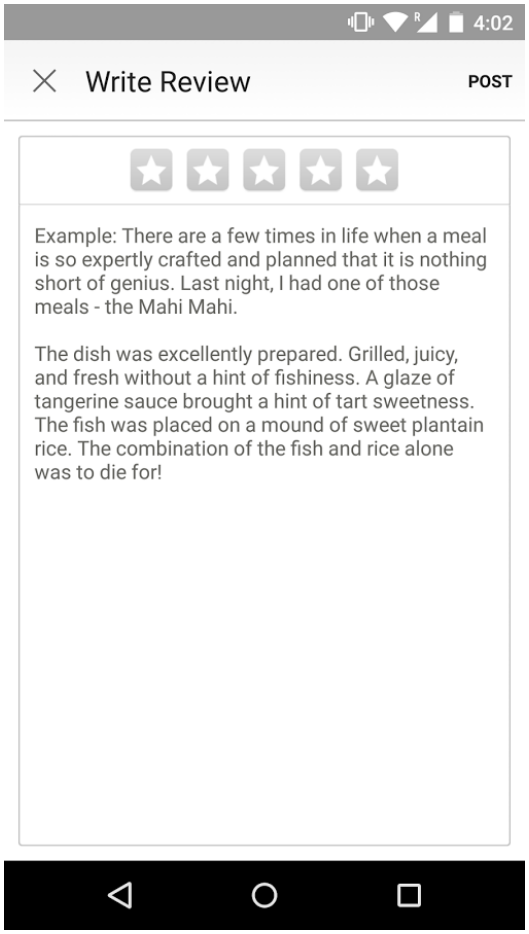
Client

Server

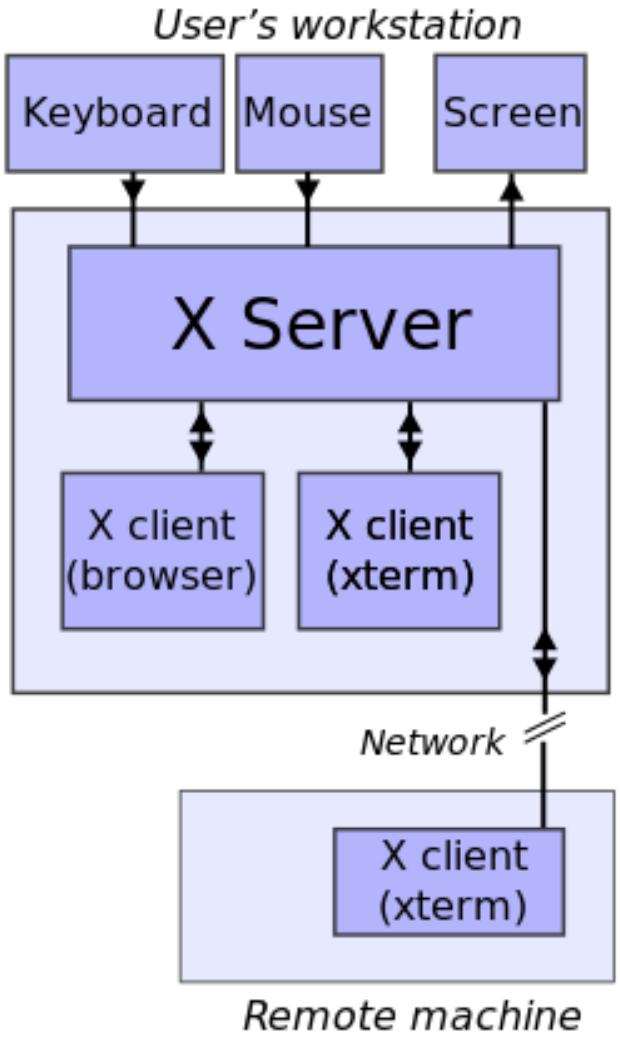
Database

Where to validate user input?

Example: Yelp App

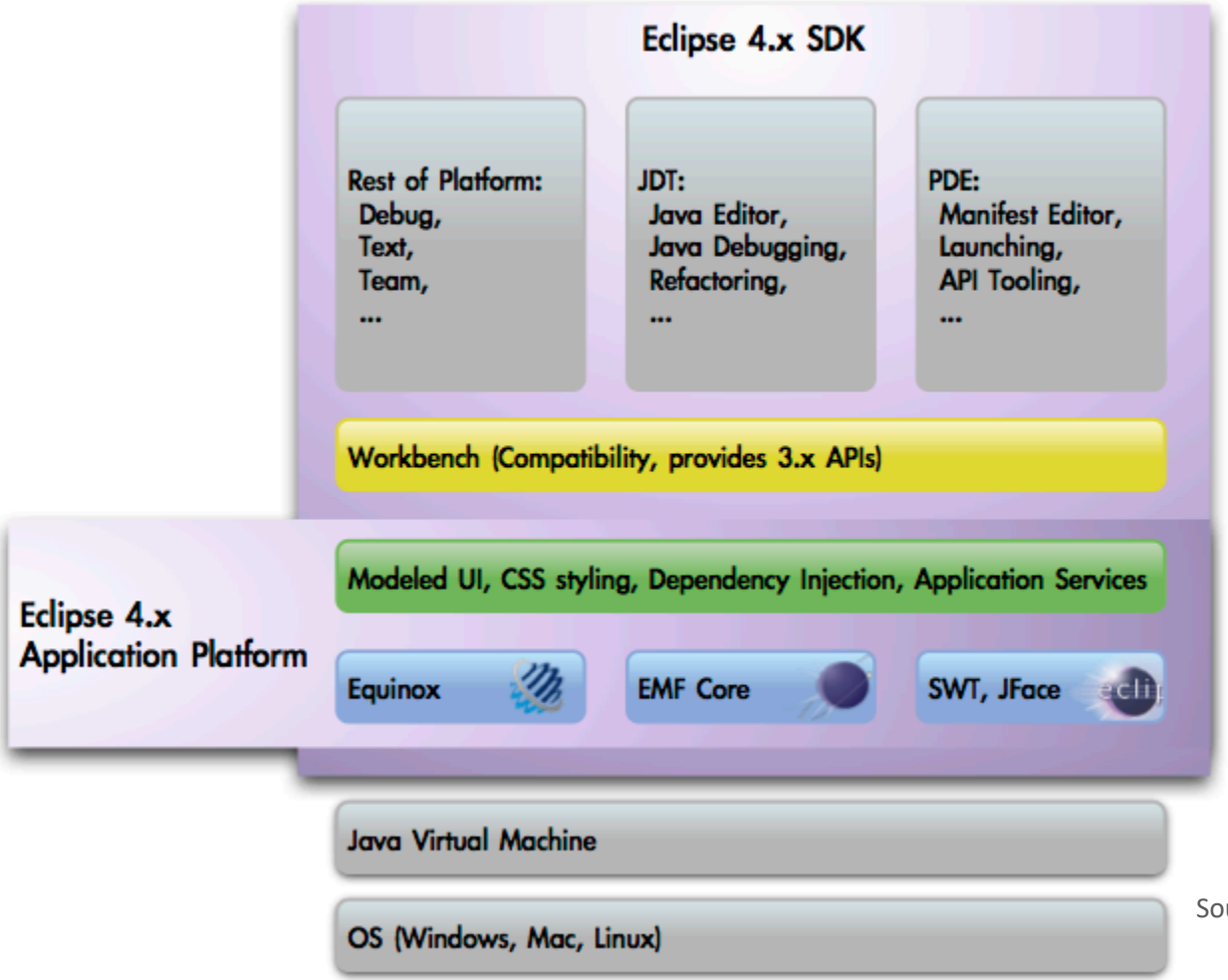


Client-server style



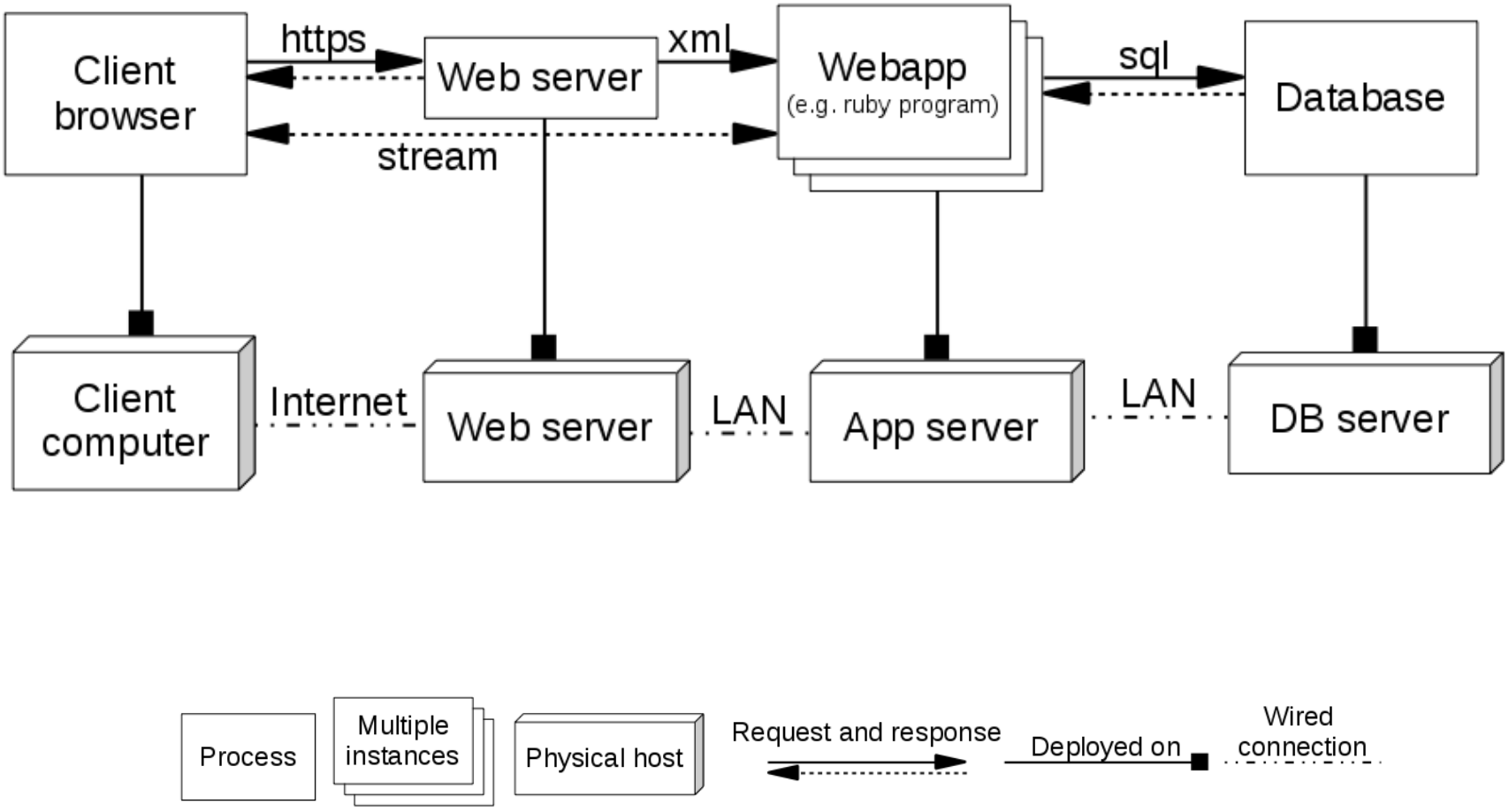
Source: wikimedia commons

Layered system



Source: eclipse.org

Tiered architecture



Architectural Style?

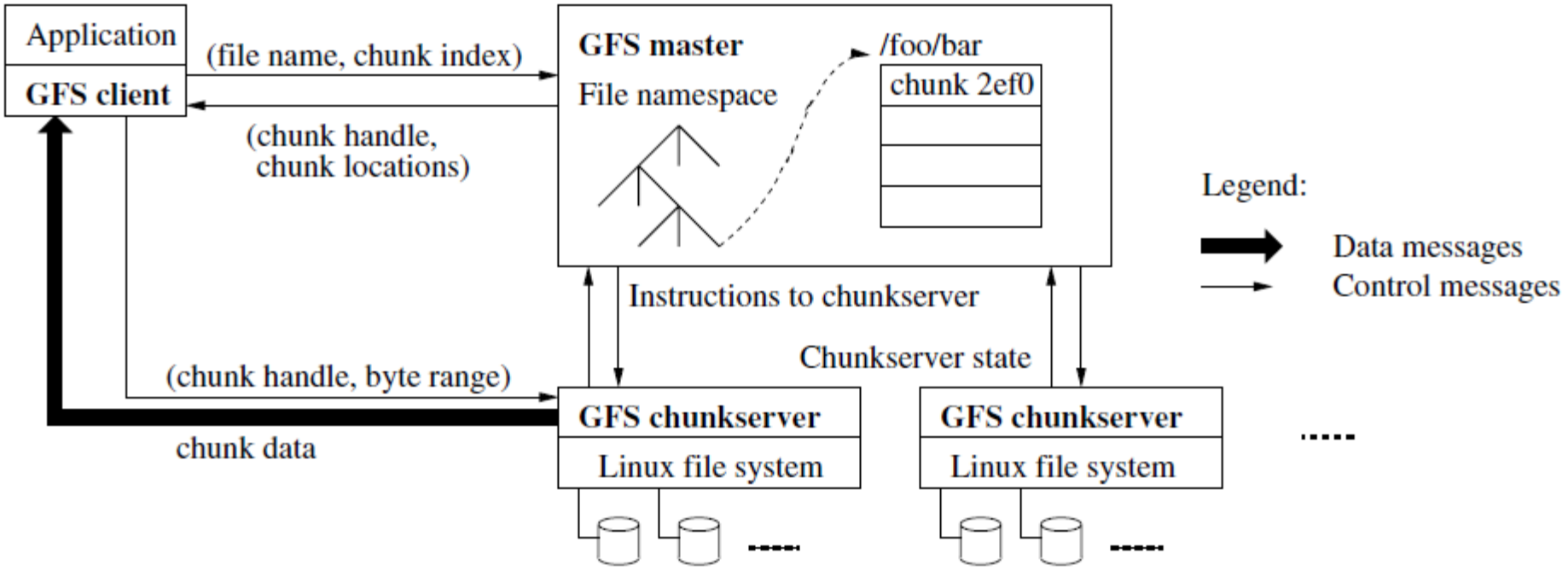


Figure 1: GFS Architecture

Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." *ACM SIGOPS operating systems review*. Vol. 37. No. 5. ACM, 2003.

Tactics

- Architectural techniques to achieve qualities
  - More tied to specific context and quality
- Smaller scope than architectural patterns
  - Problem solved by patterns: “How do I structure my (sub)system?”
  - Problem solved by tactics: “How do I get better at quality X?”
- Collection of common strategies and known solutions
  - Resemble OO design patterns

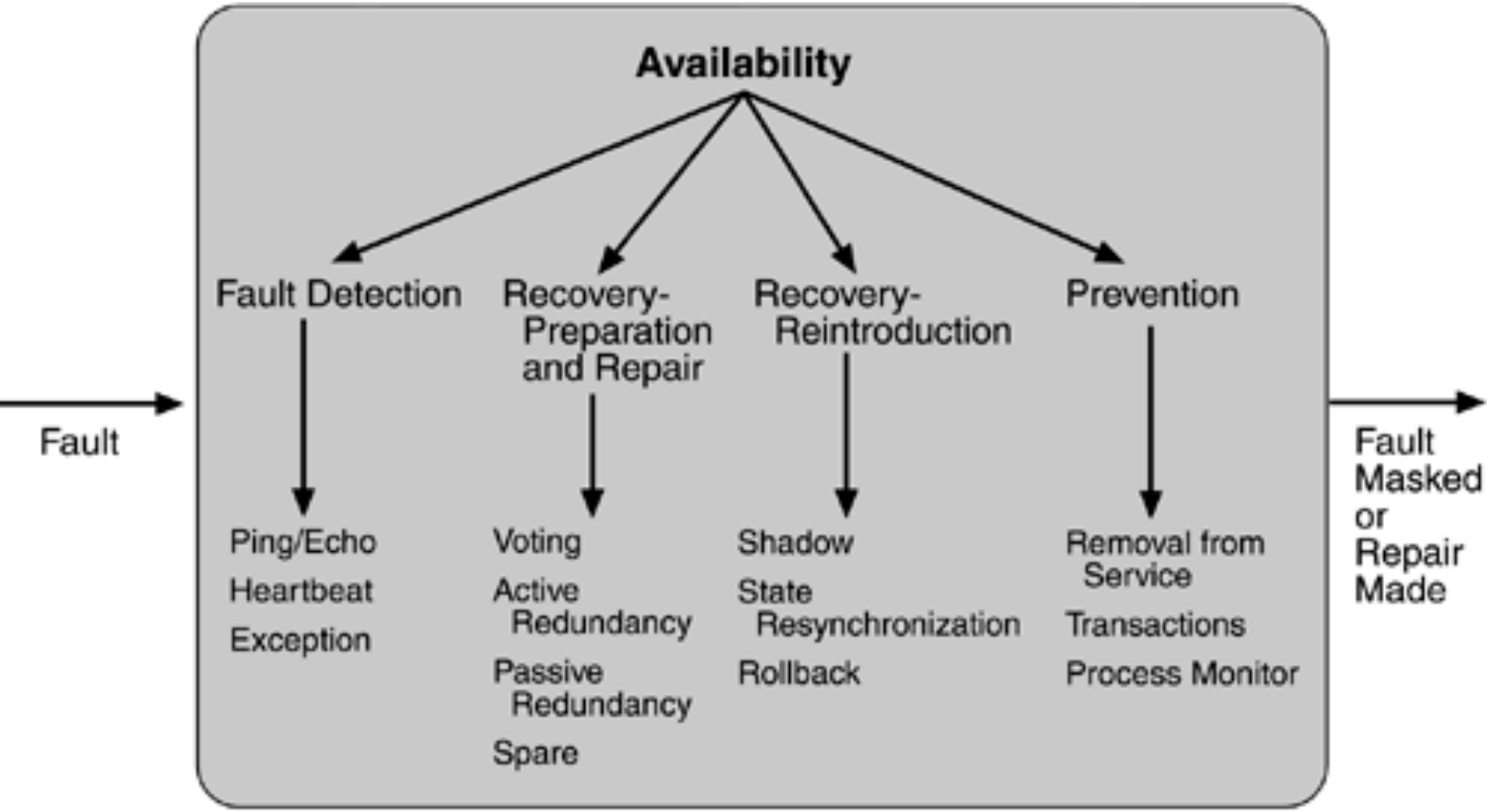

Prioritization
Concurrency
Nondeterminism
Redundancy
Authorization
Synchronization
Voting
Encryption
Separation
Authentication
Transactions
Cohesion
Sandboxing
Undo
Coupling
Timestamps
Auditing
Sanitation

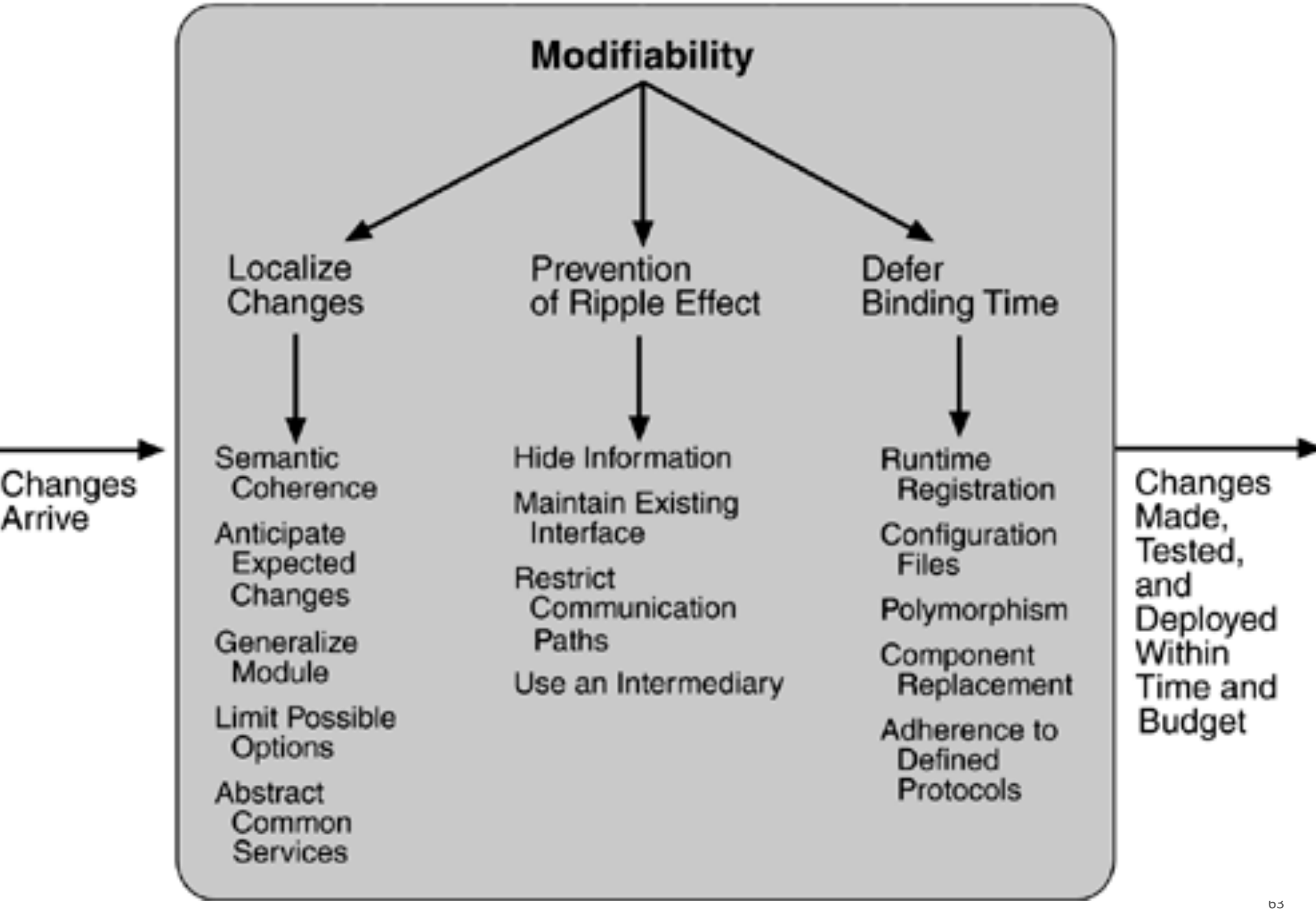
Example Tactic Description: Record/playback

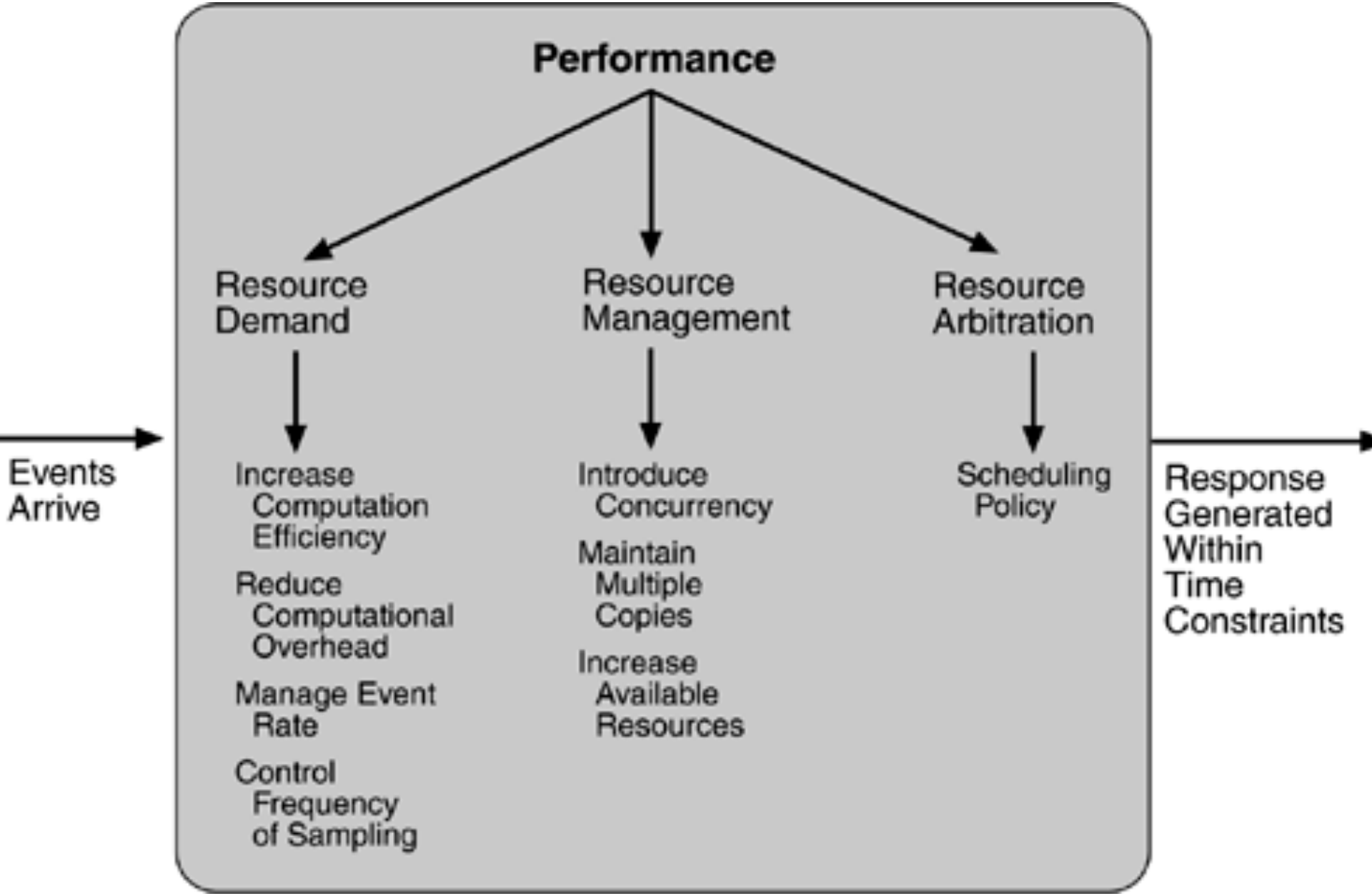
- Record/playback refers to both capturing information crossing an interface and using it as input into the test harness. The information crossing an interface during normal operation is saved in some repository and represents output from one component and input to another. Recording this information allows test input for one of the components to be generated and test output for later comparison to be saved.

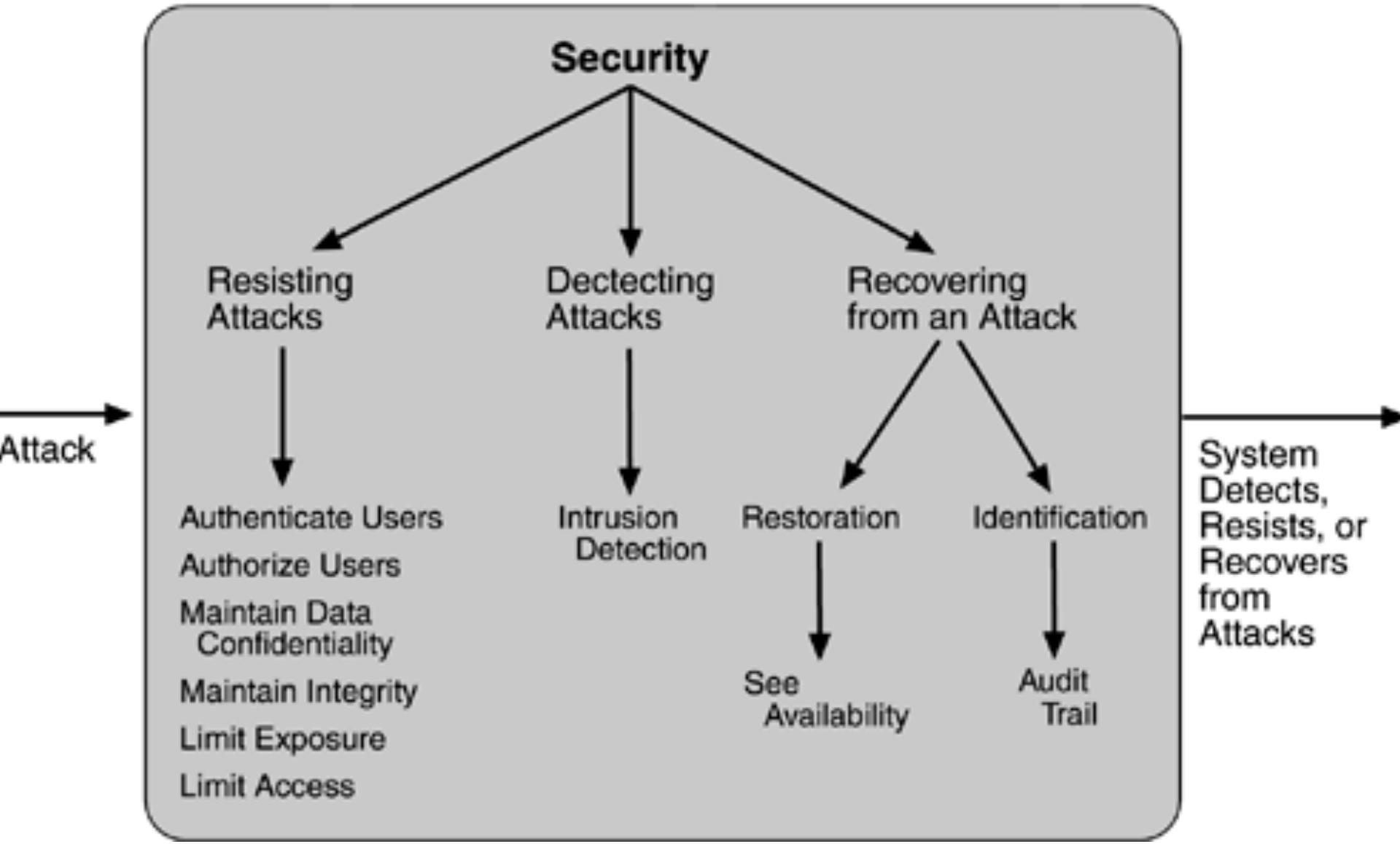
Example Tactic Description: Built-in monitors

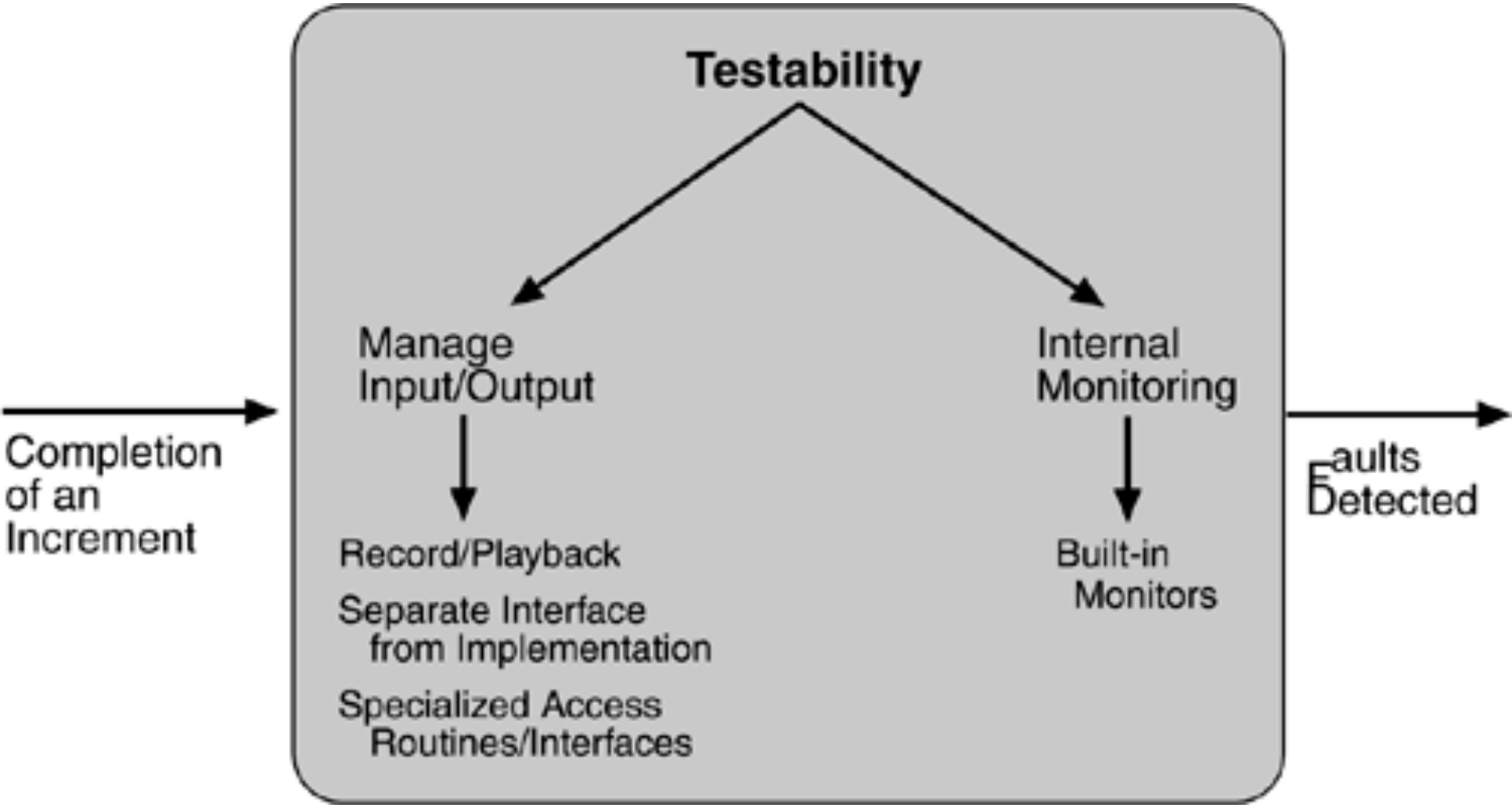
- The component can maintain state, performance load, capacity, security, or other information accessible through an interface. This interface can be a permanent interface of the component or it can be introduced temporarily via an instrumentation technique such as aspect-oriented programming or preprocessor macros. A common technique is to record events when monitoring states have been activated. Monitoring states can actually increase the testing effort since tests may have to be repeated with the monitoring turned off. Increased visibility into the activities of the component usually more than outweigh the cost of the additional testing.

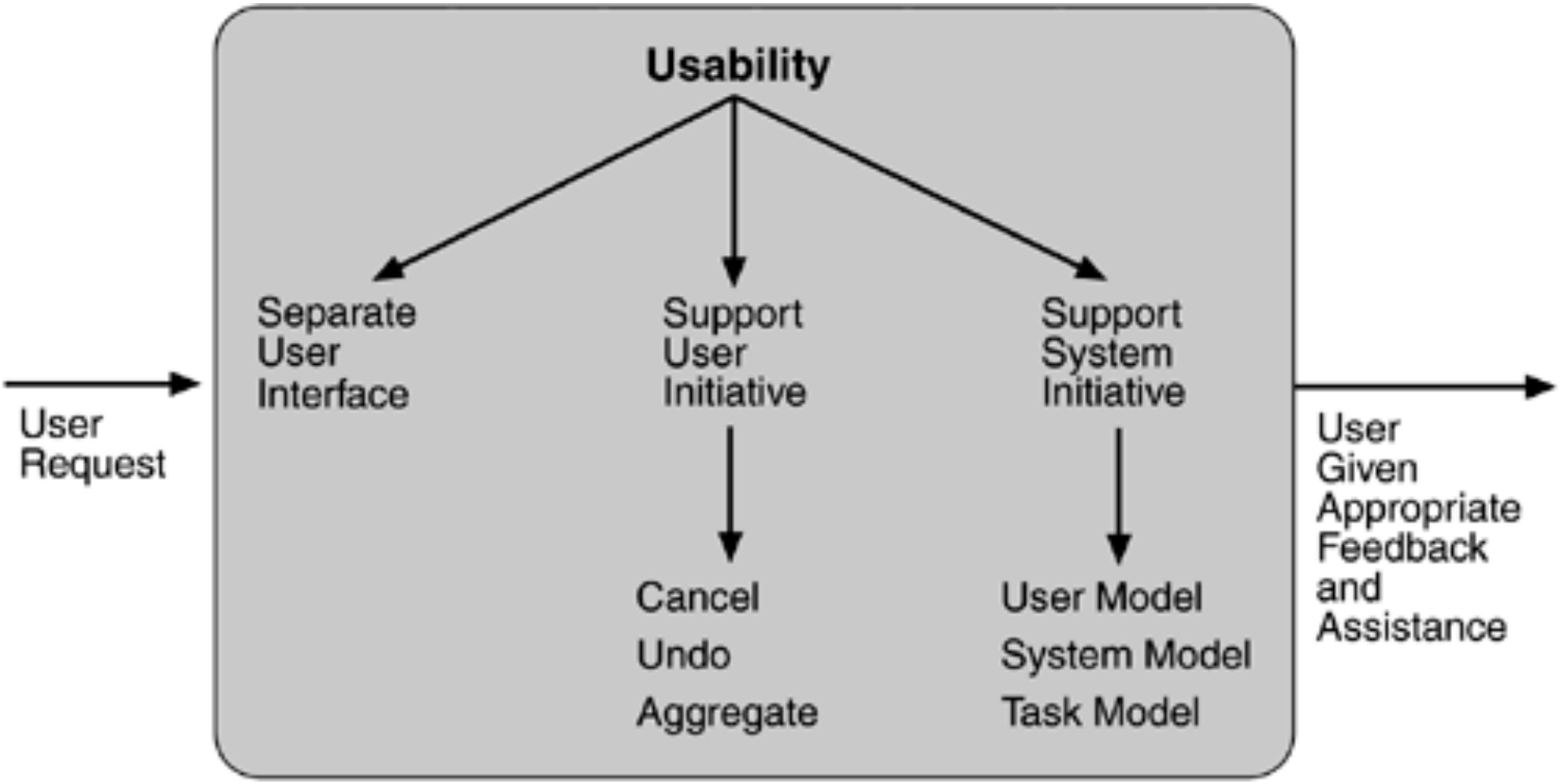

Availability
Fault Detection
Recovery-Preparation and Repair
Recovery-Reintroduction
Prevention
Fault
Ping/Echo
Heartbeat
Exception
Voting
Active Redundancy
Passive Redundancy
Spare
Shadow
State Resynchronization
Rollback
Removal from Service
Transactions
Process Monitor
Fault Masked or Repair Made

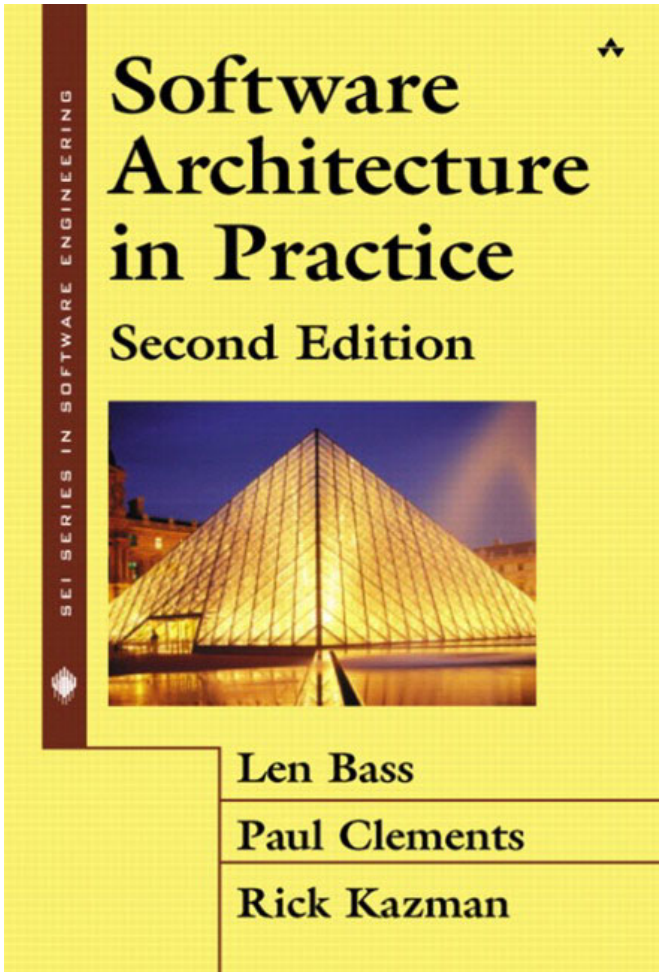

Modifiability
Localize Changes
Prevention of Ripple Effect
Defer Binding Time
Changes Arrive
Semantic Coherence
Anticipate Expected Changes
Generalize Module
Limit Possible Options
Abstract Common Services
Hide Information
Maintain Existing Interface
Restrict Communication Paths
Use an Intermediary
Runtime Registration
Configuration Files
Polymorphism
Component Replacement
Adherence to Defined Protocols
Changes Made, Tested, and Deployed Within Time and Budget


Performance
Resource Demand
Resource Management
Resource Arbitration
Events Arrive
Increase Computation Efficiency
Reduce Computational Overhead
Manage Event Rate
Control Frequency of Sampling
Introduce Concurrency
Maintain Multiple Copies
Increase Available Resources
Scheduling Policy
Response Generated Within Time Constraints


Security
Resisting Attacks
Dectecting Attacks
Recovering from an Attack
Attack
Authenticate Users
Authorize Users
Maintain Data Confidentiality
Maintain Integrity
Limit Exposure
Limit Access
Intrusion Detection
Restoration
Identification
See Availability
Audit Trail
System Detects, Resists, or Recovers from Attacks


Testability
Manage Input/Output
Internal Monitoring
Completion of an Increment
Record/Playback
Separate Interface from Implementation
Specialized Access Routines/Interfaces
Built-in Monitors
aults Detected


Usability
Separate User Interface
Support User Initiative
Support System Initiative
Cancel
Undo
Aggregate
User Model
System Model
Task Model
User Request
User Given Appropriate Feedback and Assistance



Many tactics described in Chapter 5

Brief high-level descriptions (about 1 paragraph per tactic)

Second and more detailed third edition available as ebook through CMU library.

Summary

Architecture as structures and relations
- Patterns
- Tactics

Architecture as documentation
- Views
- Rationale

Architecture as process
- Decisions
- Evaluation
- Reconstruction
- Agile

Further Readings

- Bass, Clements, and Kazman. Software Architecture in Practice. Addison-Wesley, 2003.
- Boehm and Turner. Balancing Agility and Discipline: A Guide for the Perplexed, 2003.
- Clements, Bachmann, Bass, Garlan, Ivers, Little, Merson, Nord, Stafford. Documenting Software Architectures: Views and Beyond, 2010.
- Fairbanks. Just Enough Software Architecture. Marshall & Brainerd, 2010.
- Jansen and Bosch. Software Architecture as a Set of Architectural Design Decisions, WICSA 2005.
- Lattanze. Architecting Software Intensive Systems: a Practitioner's Guide, 2009.
- Sommerville. Software Engineering. Edition 7/8, Chapters 11-13
- Taylor, Medvidovic, and Dashofy. Software Architecture: Foundations, Theory, and Practice. Wiley, 2009.