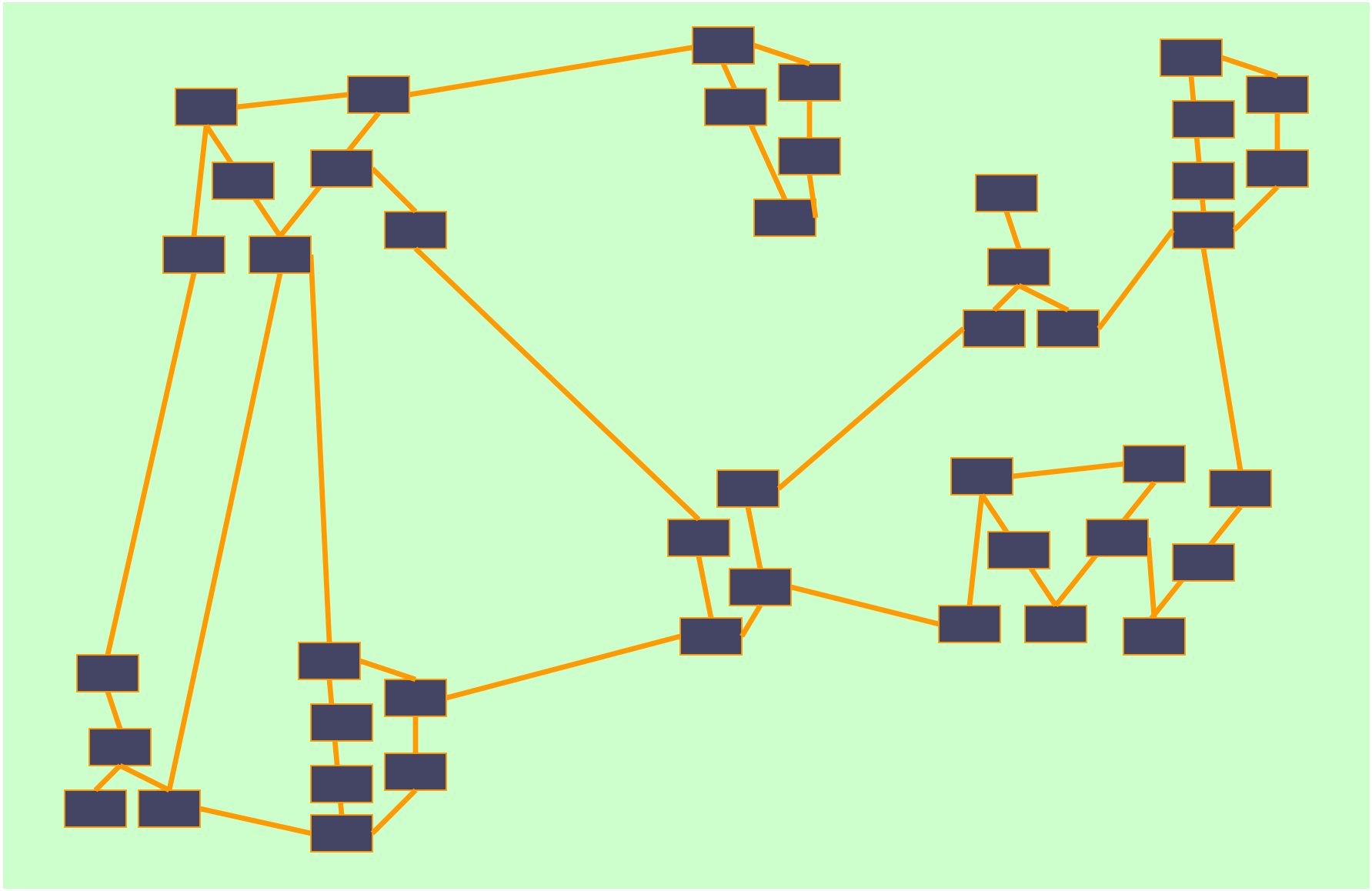
Toward a Framework for Classifying Disconnected Operation Techniques

Marija Mikic-Rakic
Nenad Medvidovic
University of Southern California
{marija, neno}@usc.edu

Motivation



Approach

- System monitoring
- Estimation of optimal deployment architecture
  - Exponentially complex problem
- Effecting the redeployment architecture

• Has this problem been solved?
• What are other disconnected operation techniques, besides redeployment?
• Can I combine them with my approach?

Disconnected operation techniques

- Caching
- Hoarding
- Queuing of remote procedure calls
- Deployment and redeployment
- Replica reconciliation
- Code mobility

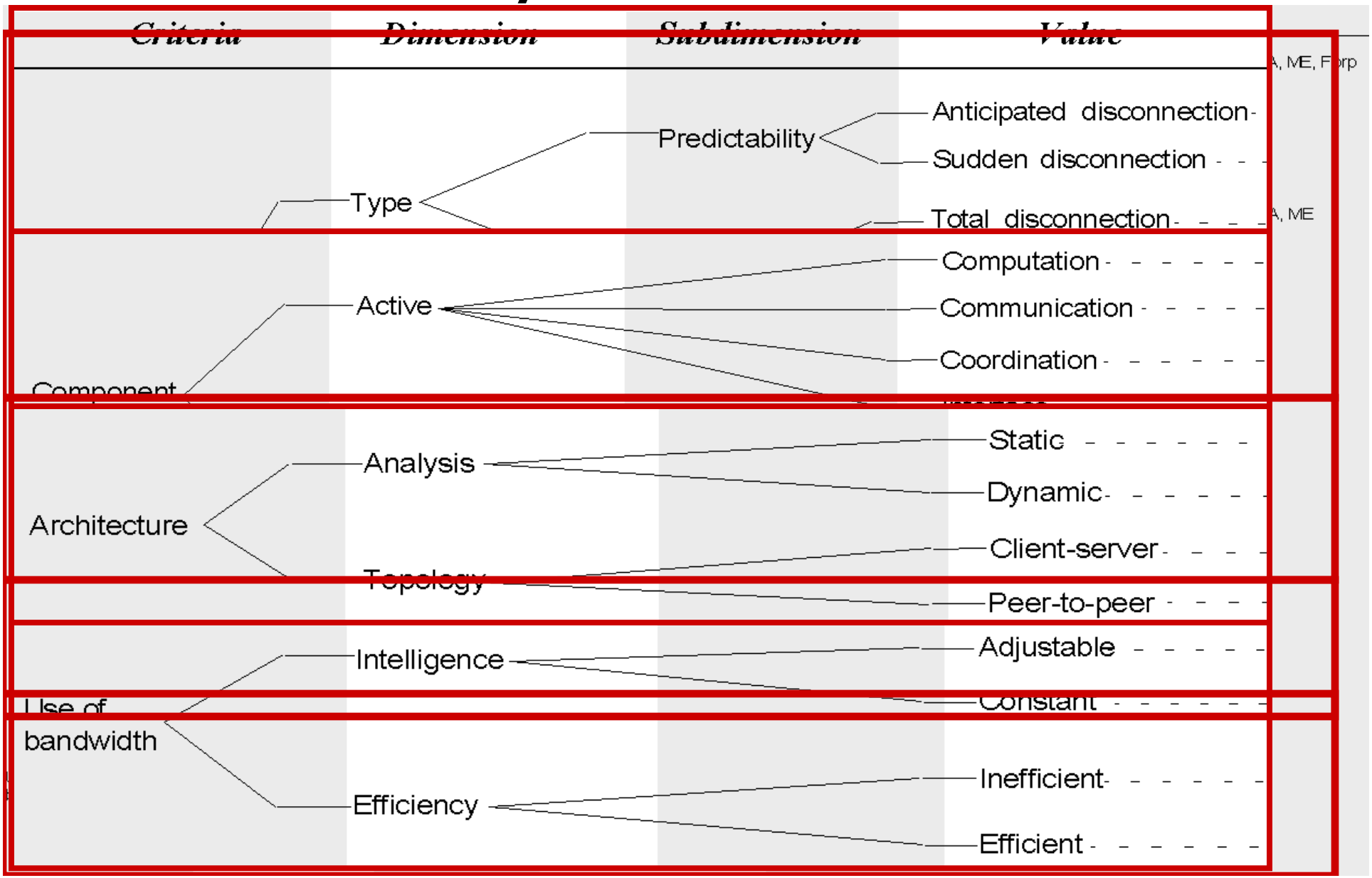
This does not tell me:
- Under which conditions can I apply a given technique
- Which techniques are (in)compatible

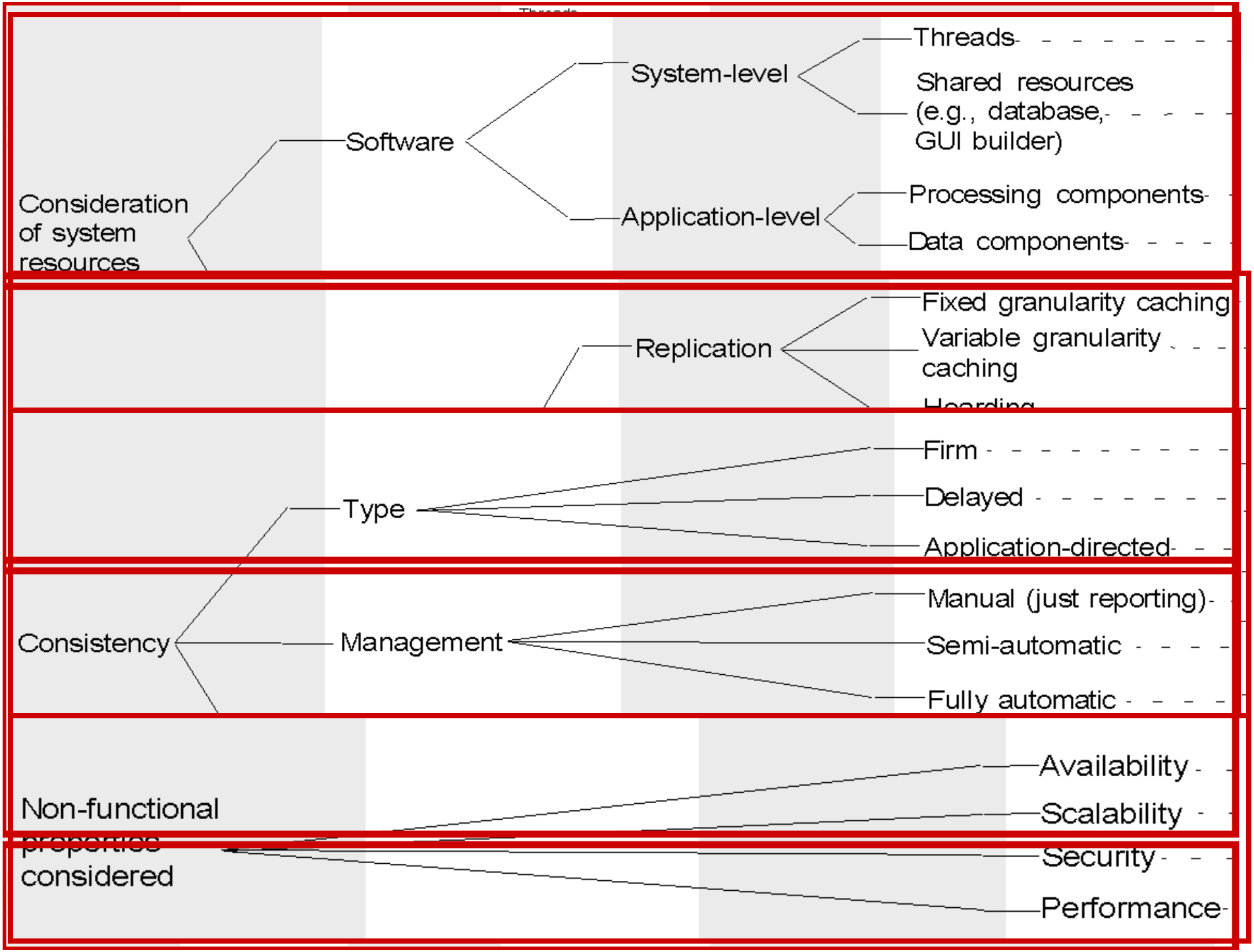
Existing approaches

- Distributed file systems
  (Coda, Ficus, D-NFS, PFS)
- Distributed databases
  (Thor, Bayou)
- Code mobility
  (Rover, Jamp, Mobile Extensions (ME), Odyssey, FarGo-DA)
- Ad-hoc networking
  (Forp, PCP, Monarch)

This does not tell me:
• Which approach works better for problem at hand
• Can I combine approach A and approach B

Taxonomy


Criteria
Dimension
Subdimension
Value
Component
Type
Predictability
Anticipated disconnection
Sudden disconnection
Total disconnection
Active
Computation
Communication
Coordination
Architecture
Analysis
Static
Dynamic
Topology
Client-server
Peer-to-peer
Use of bandwidth
Intelligence
Adjustable
Constant
Efficiency
Inefficient
Efficient


Consideration of system resources
Software
System-level
Threads
Shared resources (e.g., database, GUI builder)
Application-level
Processing components
Data components
Replication
Fixed granularity caching
Variable granularity caching
Hoarding
Consistency
Type
Firm
Delayed
Application-directed
Management
Manual (just reporting)
Semi-automatic
Fully automatic
Non-functional properties considered
Availability
Scalability
Security
Performance

Assessment of existing approaches

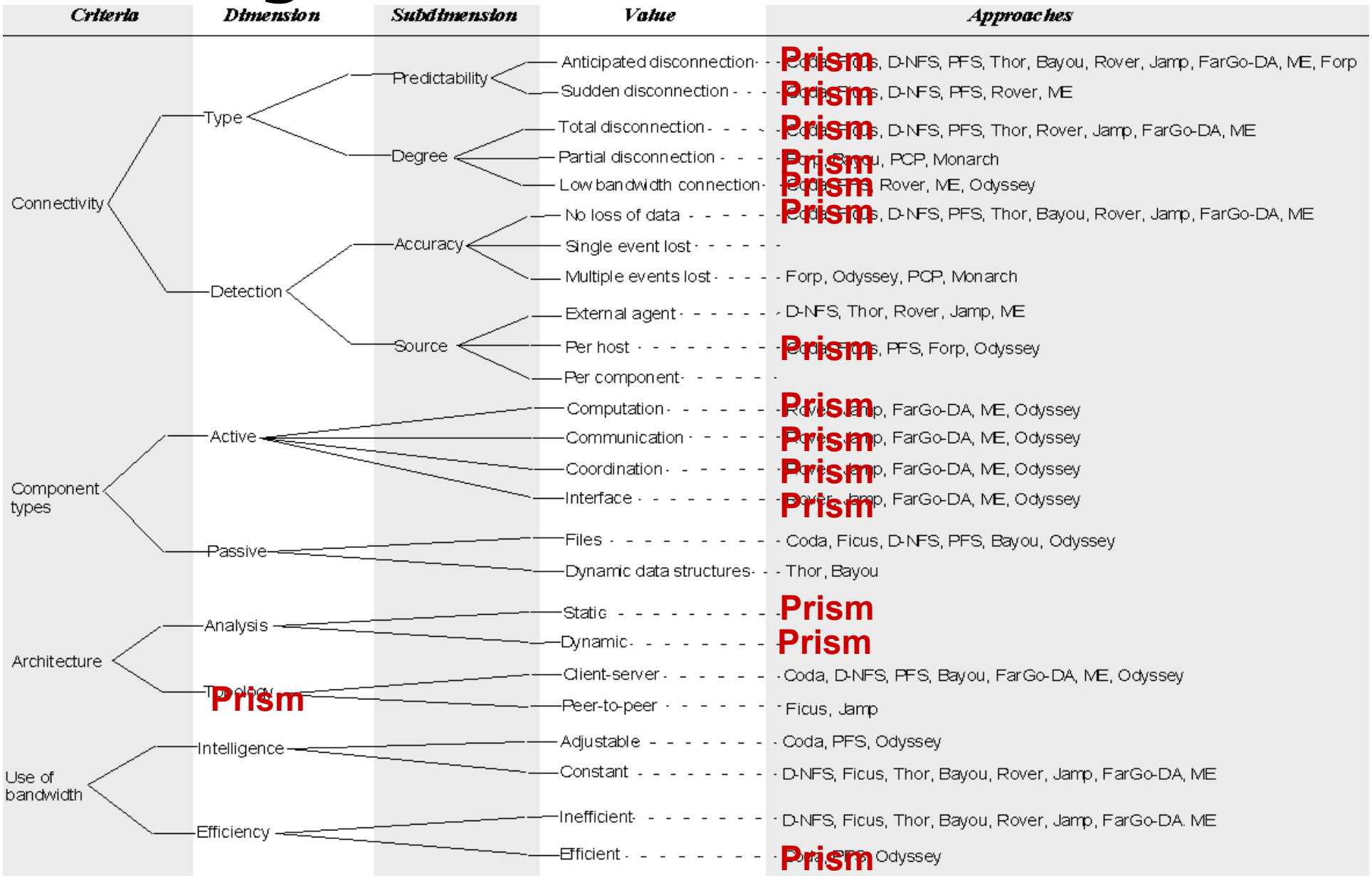
- Most focus on anticipated disconnection, maximizing availability
- Use of bandwidth
  - Intelligent and efficient (Coda, PFS, Odyssey)
  - Others assume either fully connected or disconnected mode
- System resources
  - Memory (Fargo-DA)
  - Other resources (ME and Odyssey)
- Technique
  - Application-level (Fargo-DA, ME, Odyssey)
  - Others operate at system-level
  - Most commonly used – some form of replication
  - None employ re-routing

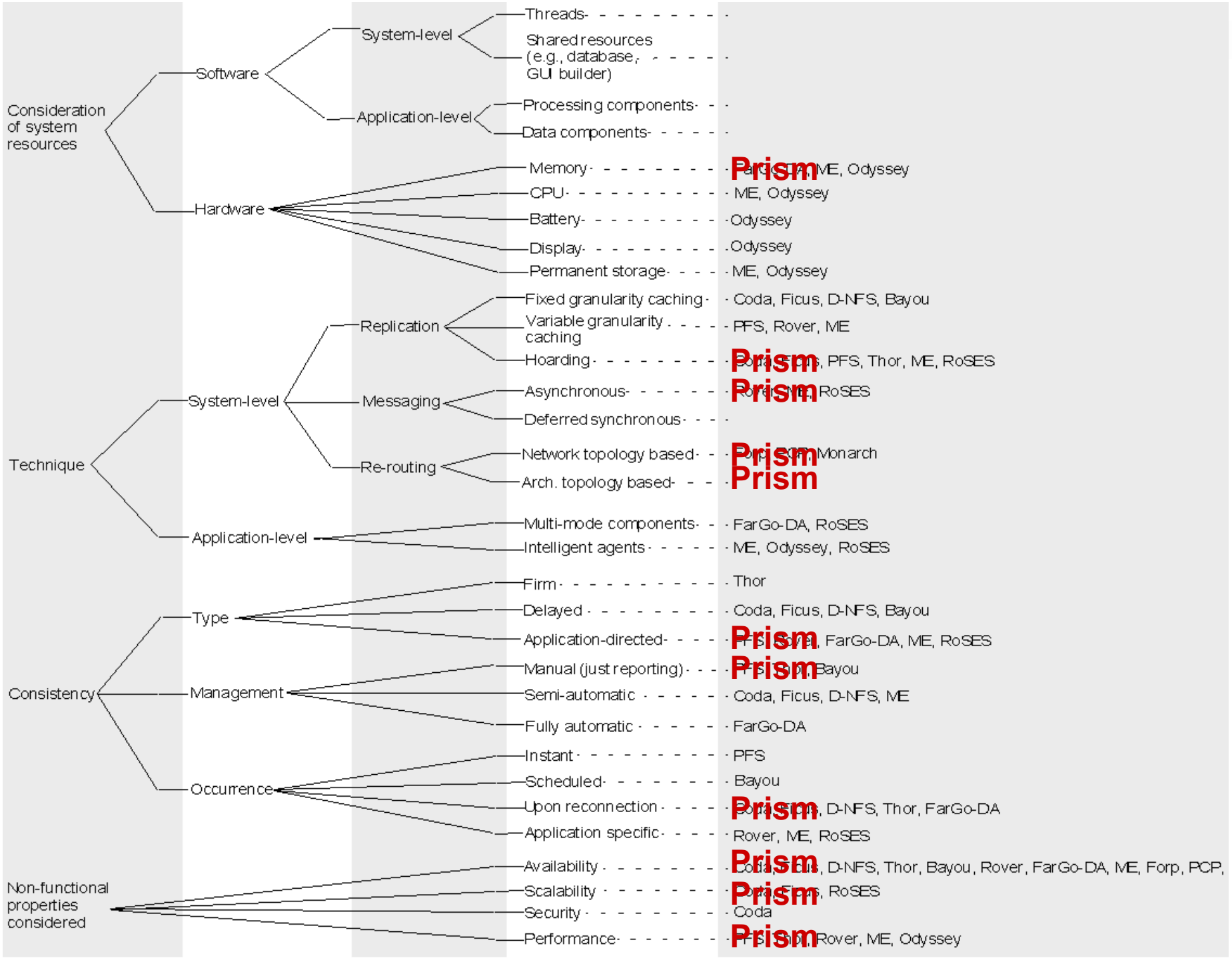
Conclusions and future work

- Understand the (in)compatibilities among the existing techniques , different dimensions, subdimensions and values
- Suggest the best possible approach or combination of approaches for the problem at hand
- Highlight the areas not currently supported
  - Suggest a research agenda

Questions?

Target of Prism research




Consideration of system resources
Software
System-level
Threads
Shared resources (e.g., database, GUI builder)
Application-level
Processing components
Data components
Hardware
Memory
CPU
Battery
Display
Permanent storage
ME, Odyssey
ME, Odyssey
Odyssey
Odyssey
ME, Odyssey
Technique
System-level
Replication
Fixed granularity caching
Variable granularity caching
Hoarding
Coda, Ficus, D-NFS, Bayou
PFS, Rover, ME
Coda, Ficus, PFS, Thor, ME, RoSES
Messaging
Asynchronous
Deferred synchronous
Rover, ME, RoSES
Re-routing
Network topology based
Arch. topology based
Application-level
Multi-mode components
Intelligent agents
FarGo-DA, RoSES
ME, Odyssey, RoSES
Consistency
Type
Firm
Delayed
Application-directed
Thor
Coda, Ficus, D-NFS, Bayou
PFS, Rover, FarGo-DA, ME, RoSES
Management
Manual (just reporting)
Semi-automatic
Fully automatic
Coda, Ficus, D-NFS, ME
FarGo-DA
Occurrence
Instant
Scheduled
Upon reconnection
Application specific
PFS
Bayou
Coda, Ficus, D-NFS, Thor, FarGo-DA
Rover, ME, RoSES
Non-functional properties considered
Availability
Scalability
Security
Performance
Coda, Ficus, D-NFS, Thor, Bayou, Rover, FarGo-DA, ME, Forp, PCP,
Coda, Ficus, RoSES
Coda
PFS, Bayou, Rover, ME, Odyssey
Prism
Prism
Prism
Prism
Prism
Prism
Prism
Prism
Prism