Carnegie Mellon

COASTE Center on Architecting Socio-Technical Ecosystems

Architectural Knowledge and Organizational Context: The Case for Socio-Technical Styles

James Herbsleb
Carnegie Mellon University
jdh@cs.cmu.edu

Carnegie Mellon

Photolithographic Alignment

- Small architectural changes have often killed successful firms
- Architecture gets embedded into organization
  - Communication paths
  - Cognitive filters
  - Problem-solving strategies

Henderson, R.M. & Clark, K.B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly*, 35 (1), pp. 9-30.

COASTE

Architectural Decisions

- Influence technical characteristics of product
- Also constrain design of organization
- Conway's Law
  - "Any organization that designs a system will inevitably produce a design whose structure is a copy of the organization's communication structure."

Conway, M.E. How do committees invent? *Datamation*, 14, 5 (1968), 28-31

Carnegie Mellon

Conway's Law Is Static

- Assumes architectural decisions all made up front, not changed
- Assumes requirements won't change
  - Requirements Δ → architecture Δ
- Assumes simple, static organizational structure
  - Ignores network structure
- Assumes implementation will conform to architectural specification

COASTE

Varieties of Change

- Key elements
  - Interfaces (uncertainty, complexity)
  - Allocation of functionality
  - Bending the rules
- How change propagates
  - Informing vs. negotiating
  - Interests at stake: degrees of "heat"
- Ability to accommodate change varies dramatically
  - Coordination capability
- **How to bring architectures and organizations into alignment?**

Carnegie Mellon

An Approach . . .

- See successful patterns that recur, e.g.,
  - Architectural styles
  - Design patterns
  - Problem frames
- A way of capturing knowledge for reuse
- Can we expand such ideas from technical to socio-technical?

COASTE

Architectural Style: Pipe and Filter

- Pipe-and-filter commitments, e.g.,
  - Filter performs local transformation
  - Filters are independent
  - Filters do not know identity of up- or down-stream filters
- Organizational commitments to handle change
  - Internal filter changes – local, no coordination (optimistically)
  - Interface changes – regional, just between producer/ consumer groups
  - Changes affecting global attributes – project-wide

Architectural Style: Layers

- Layered systems commitments, e.g.,
  - Each layer provides service to layer above
  - Each layer acts as a client to layer below
  - Components implement virtual machine
- Organizational commitments to handle change
  - Internal layer changes – local
  - Interface changes – local if service remains backward compatible, or client does not need new services
  - Other Interface changes – regional, only between two layers
  - Changes affecting global attributes – central, or project-wide

Socio-Technical Style

- Technical architectural style matched with organizational arrangements with the capacity to handle the kinds of coordination work the style requires.

Coordination Capacity

- People factors
  - Language skills
  - Culture
  - Expertise & TMS
  - History of collaboration
  - Organizational stability
- Organizational factors
  - Divergent incentives
  - Divergent strategies
  - Unclear goals
  - Divergent tools, practices, processes
  - Communication infrastructure

Carnegie Mellon

Coordination Capacity

- Project factors
  - Number of sites
  - Time zones
  - Disciplinary or professional boundaries
  - People have multiple teams
  - Leadership style

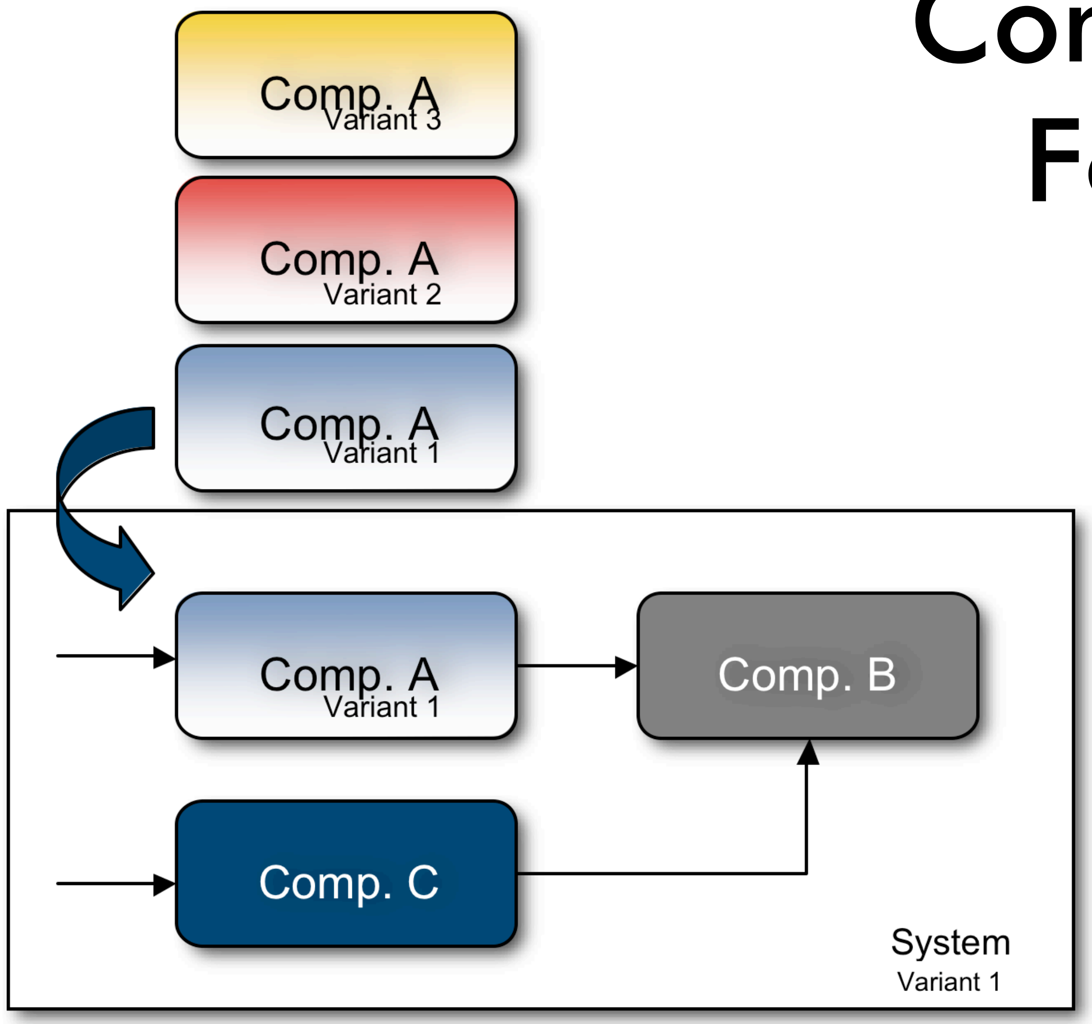
COASTE

A Few Examples . . .

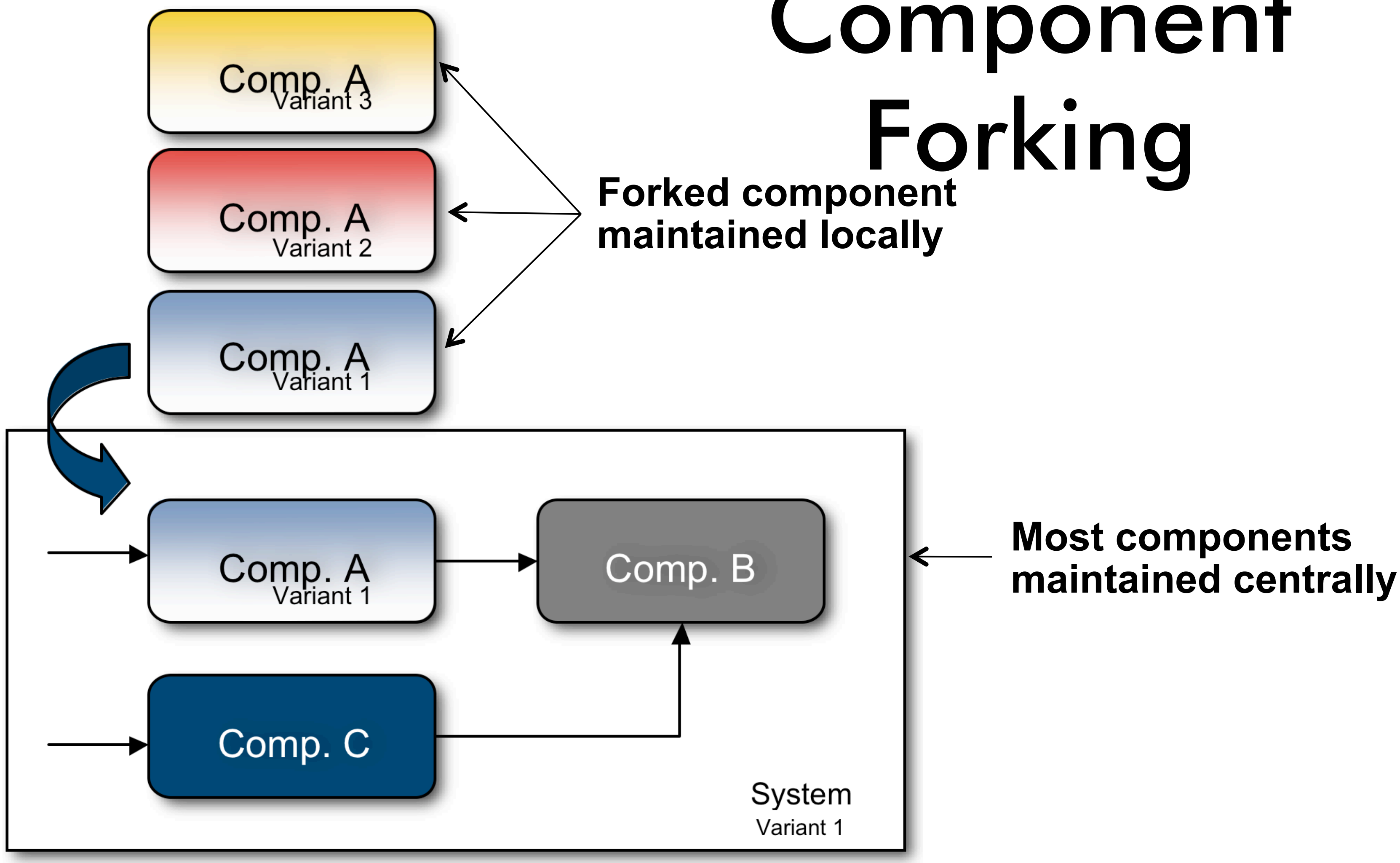
- Extracted from developers and architects at multinational engineering firm
- Idealized
- Echoes of product line engineering
- Not necessarily seen multiple times
- Have been integrated as a module in corporate training program for software architects

This work was done in collaboration with **Marcelo Cataldo** and **Sangeeth Nambiar**.

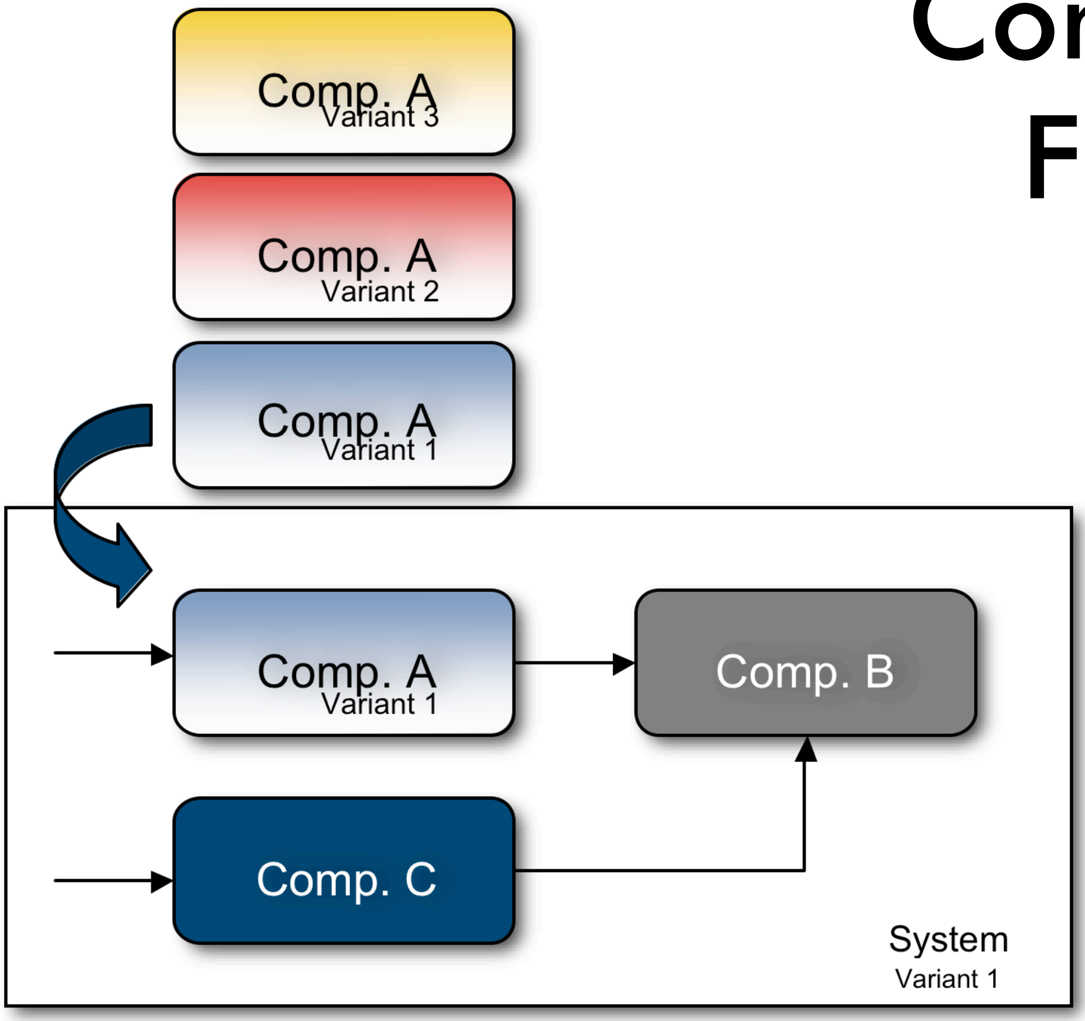
Component Forking



Component Forking



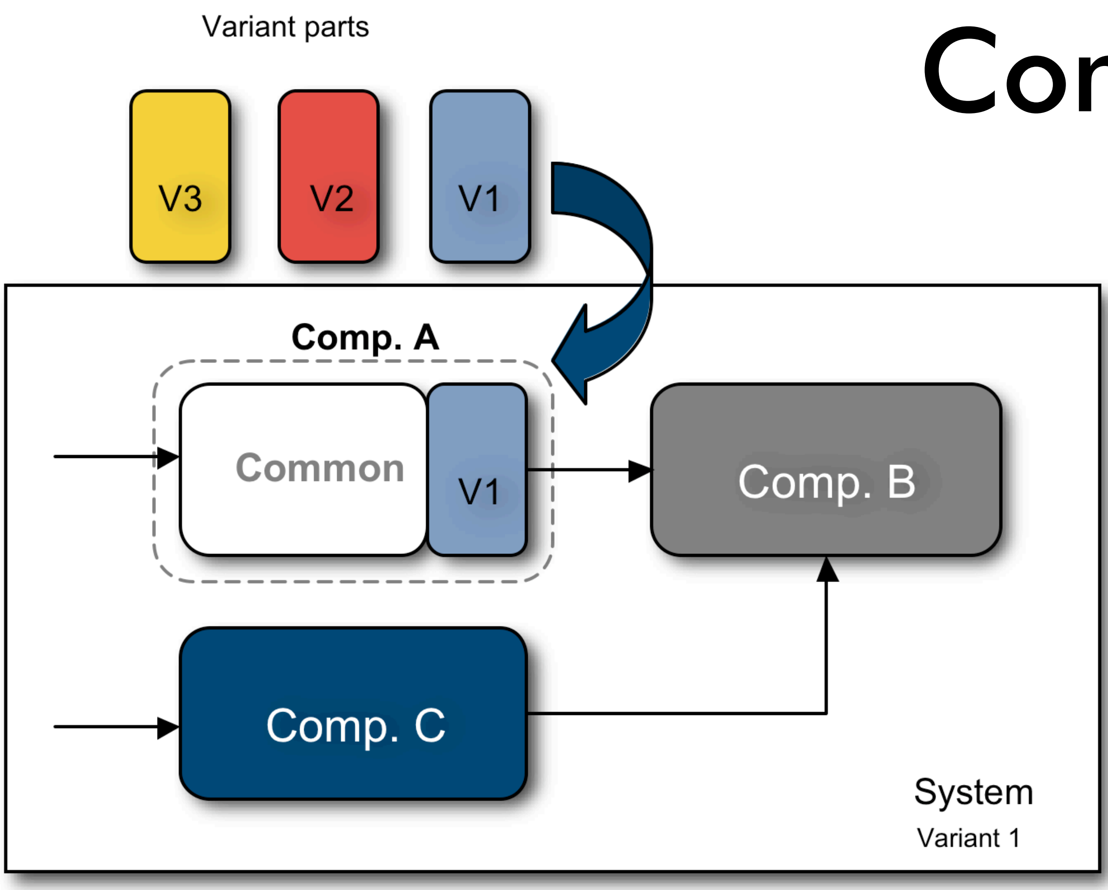
Component Forking



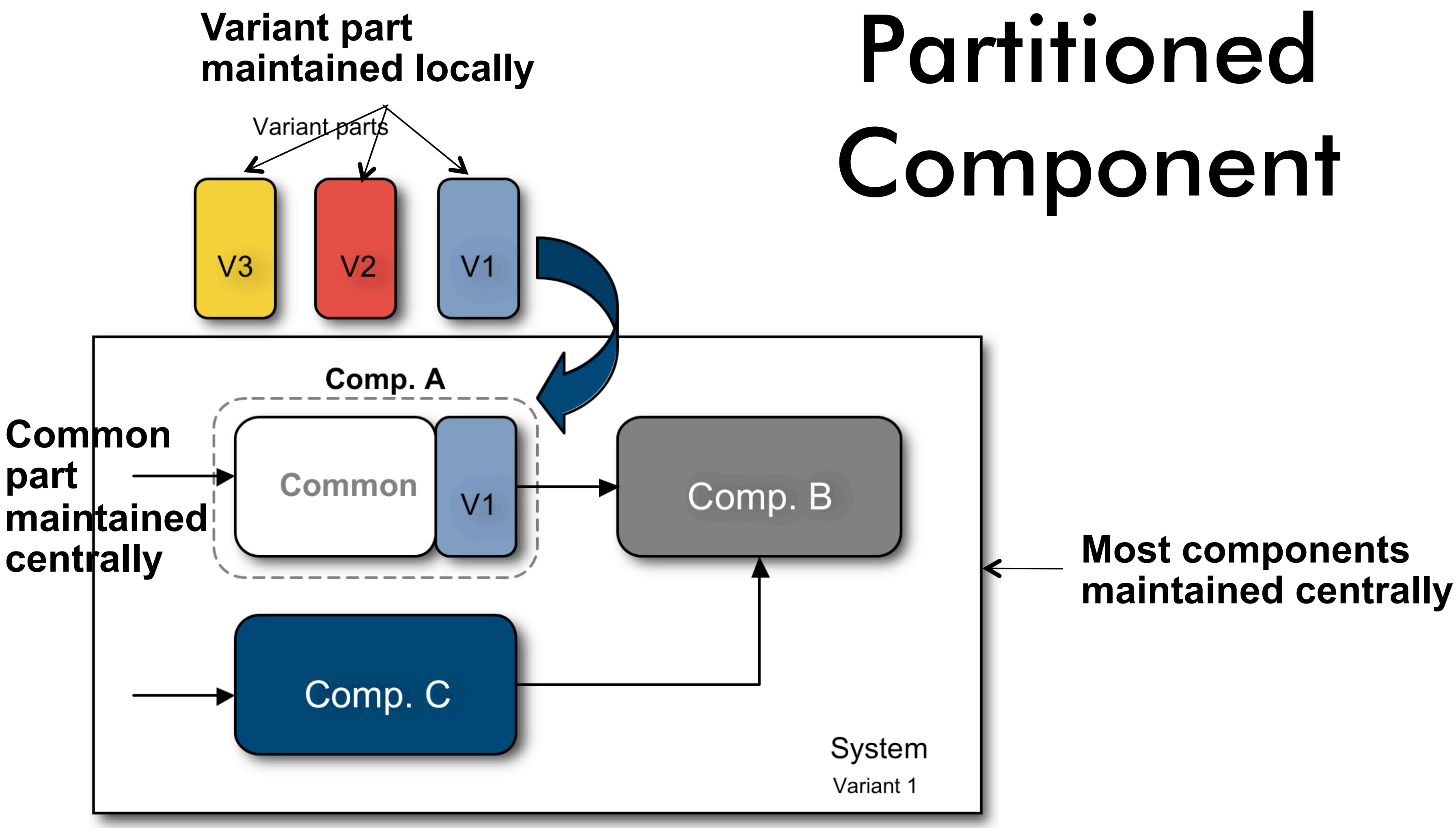
Gain: no need to coordinate across variants

Loss: duplicated effort, difficulty in maintenance, impact of changing other components difficult to anticipate

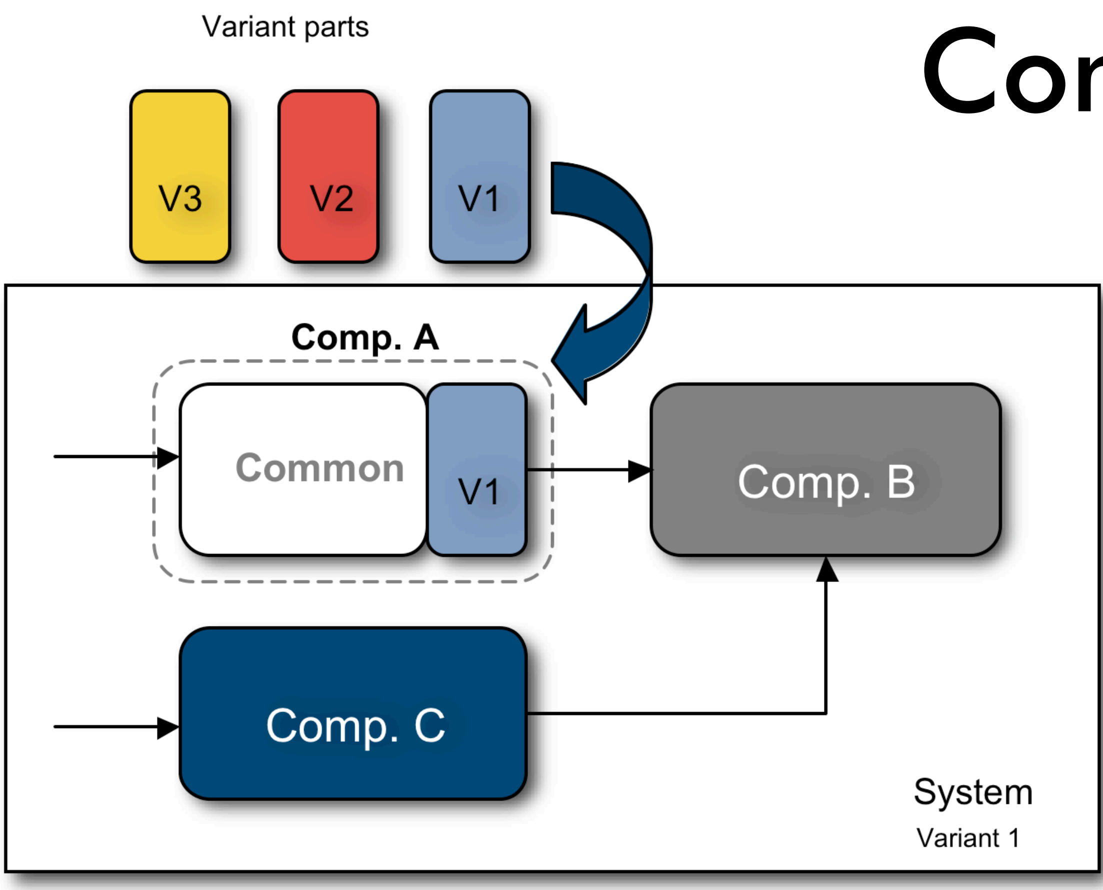
Partitioned Component



Partitioned Component



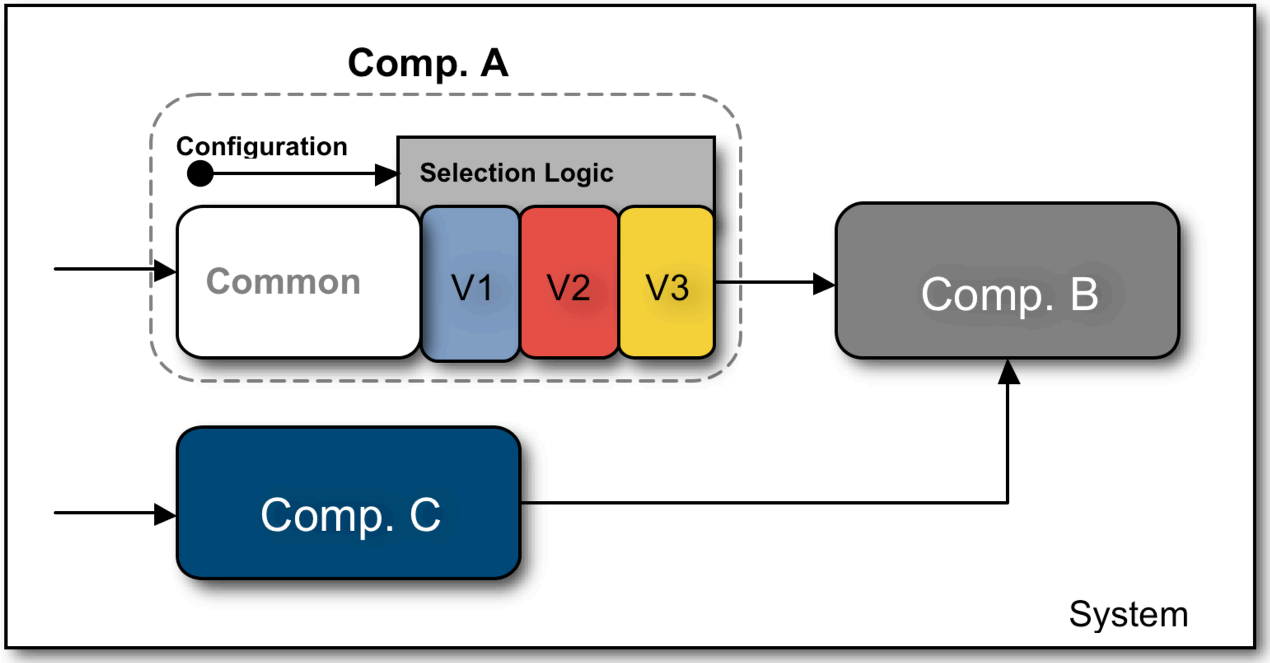
Partitioned Component



Gain: no need to coordinate across variant parts

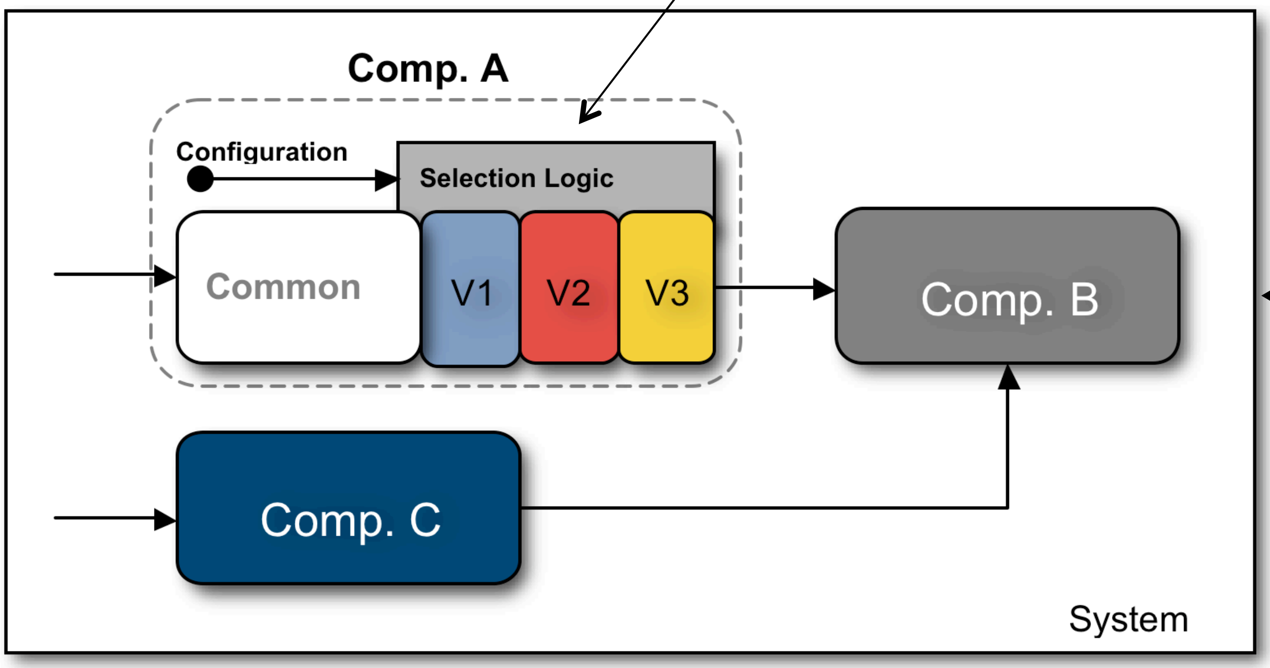
Loss: duplicated effort, difficulty in maintenance, impact of changing other components difficult to anticipate

Component Slicing



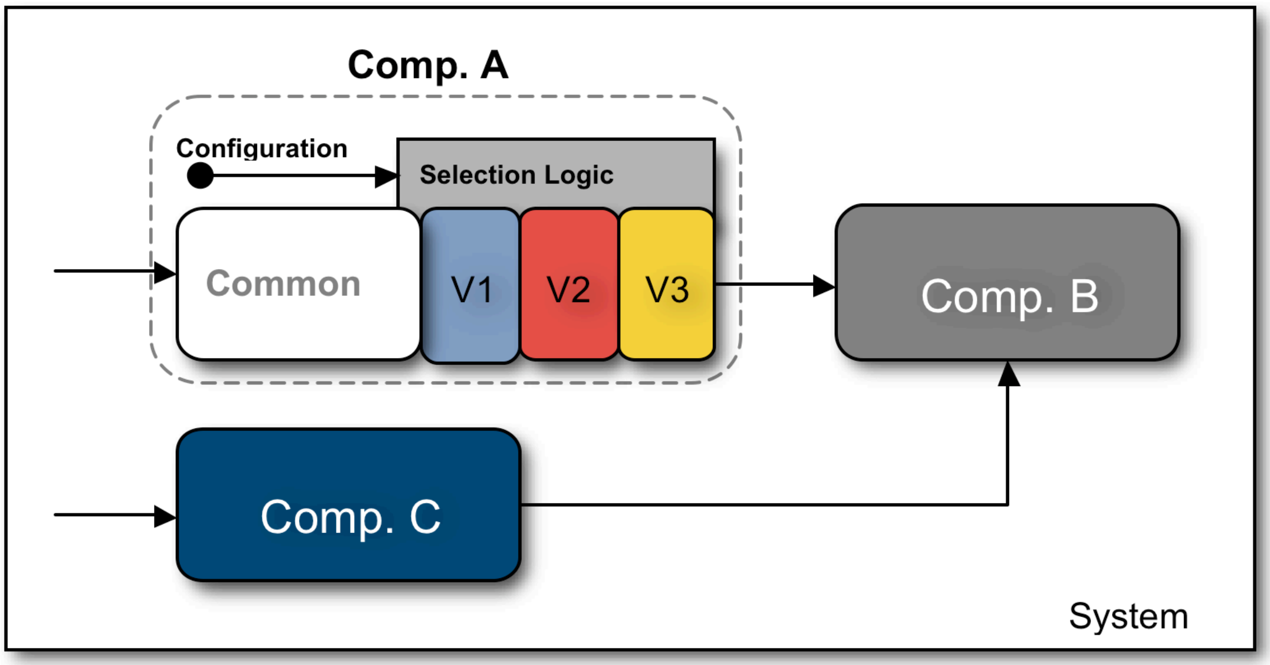
Component Slicing

Variant selected locally (configuration)



All components maintained centrally

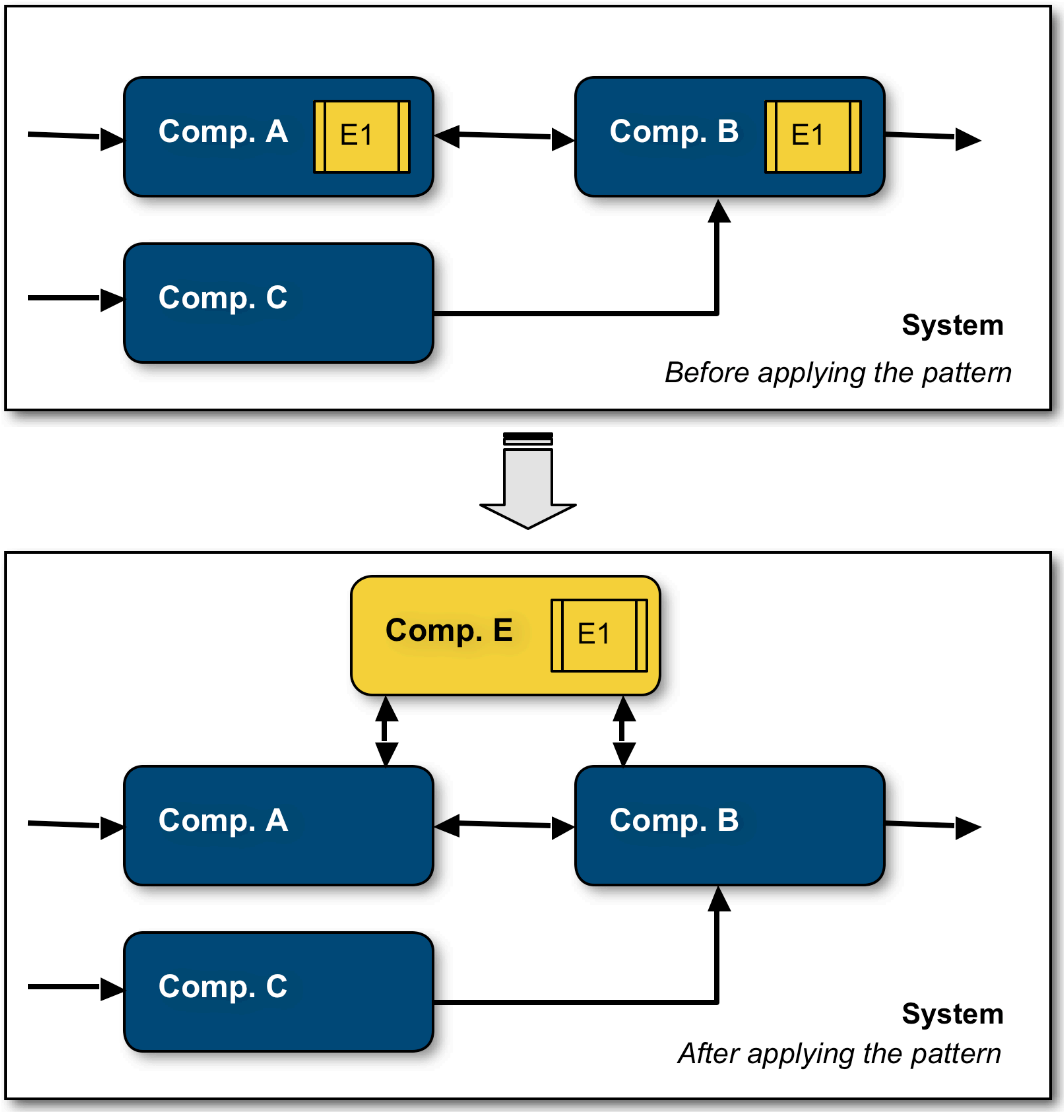
Component Slicing



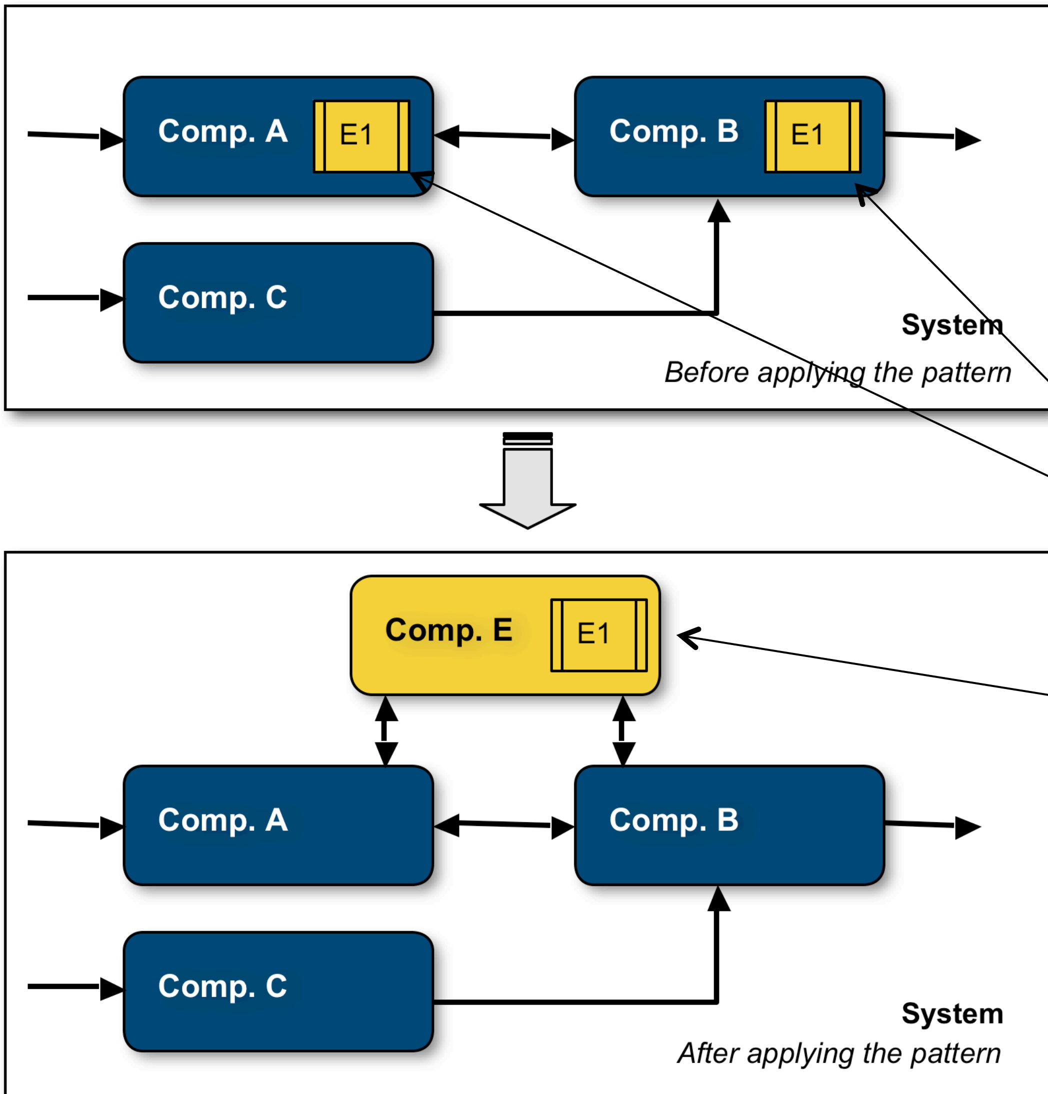
Gain: simplifies coordination around integrating and testing variants

Loss: must communicate requirements for variants to central team

Centralized Runtime Dependencies



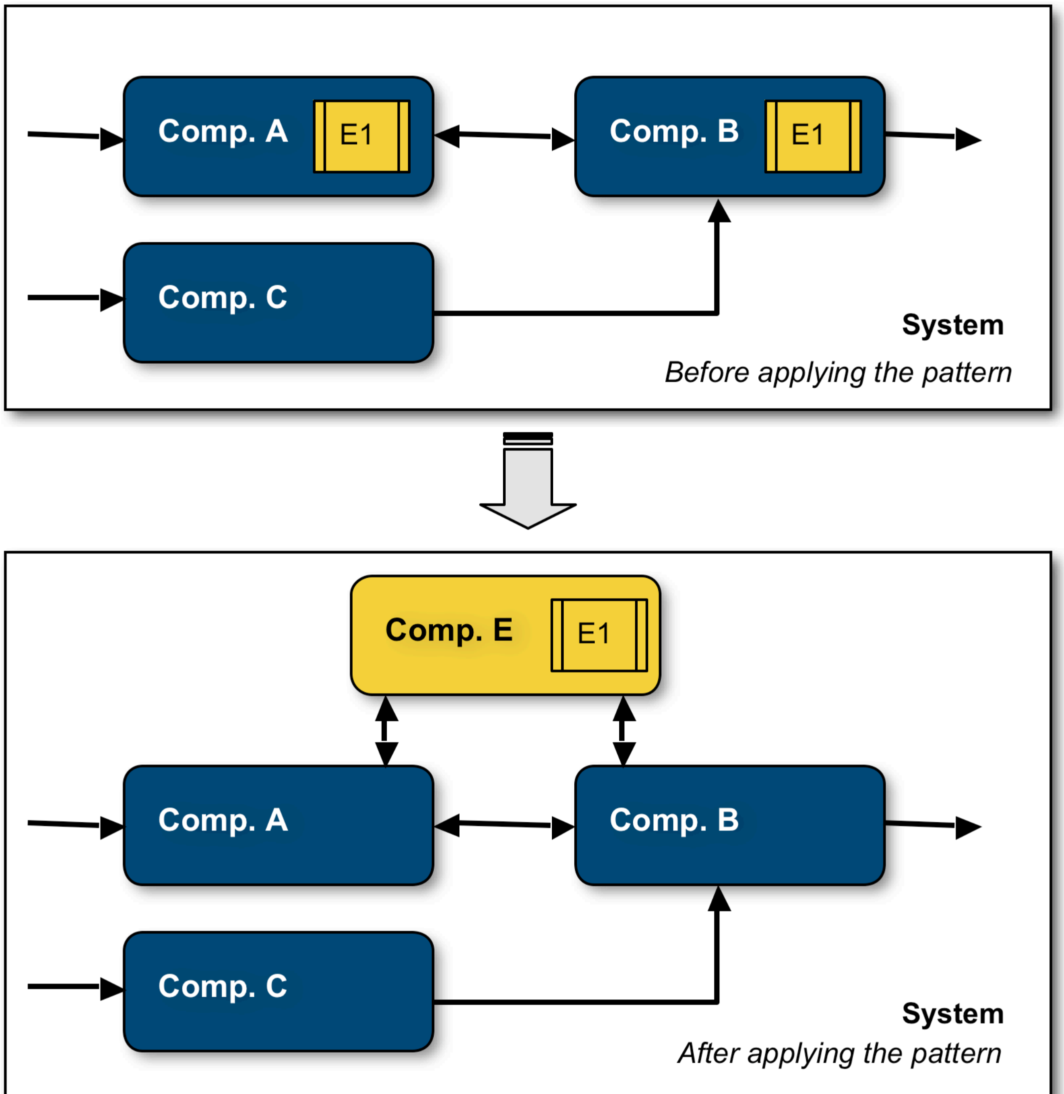
Centralized Runtime Dependencies



Runtime functionality with complex interdependencies brought together in single component

Maintained by team with global view, e.g., of error recovery

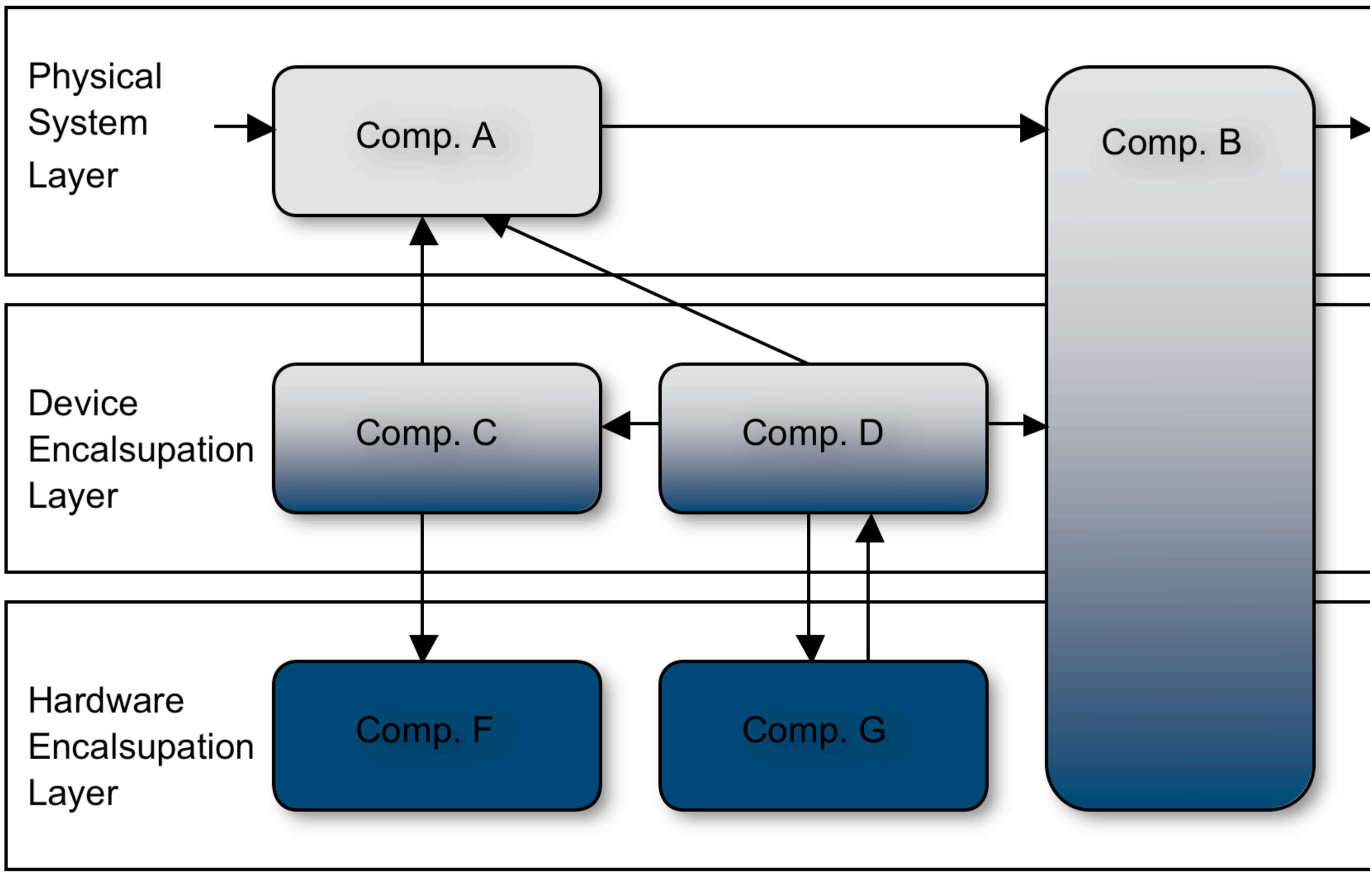
Centralized Runtime Dependencies



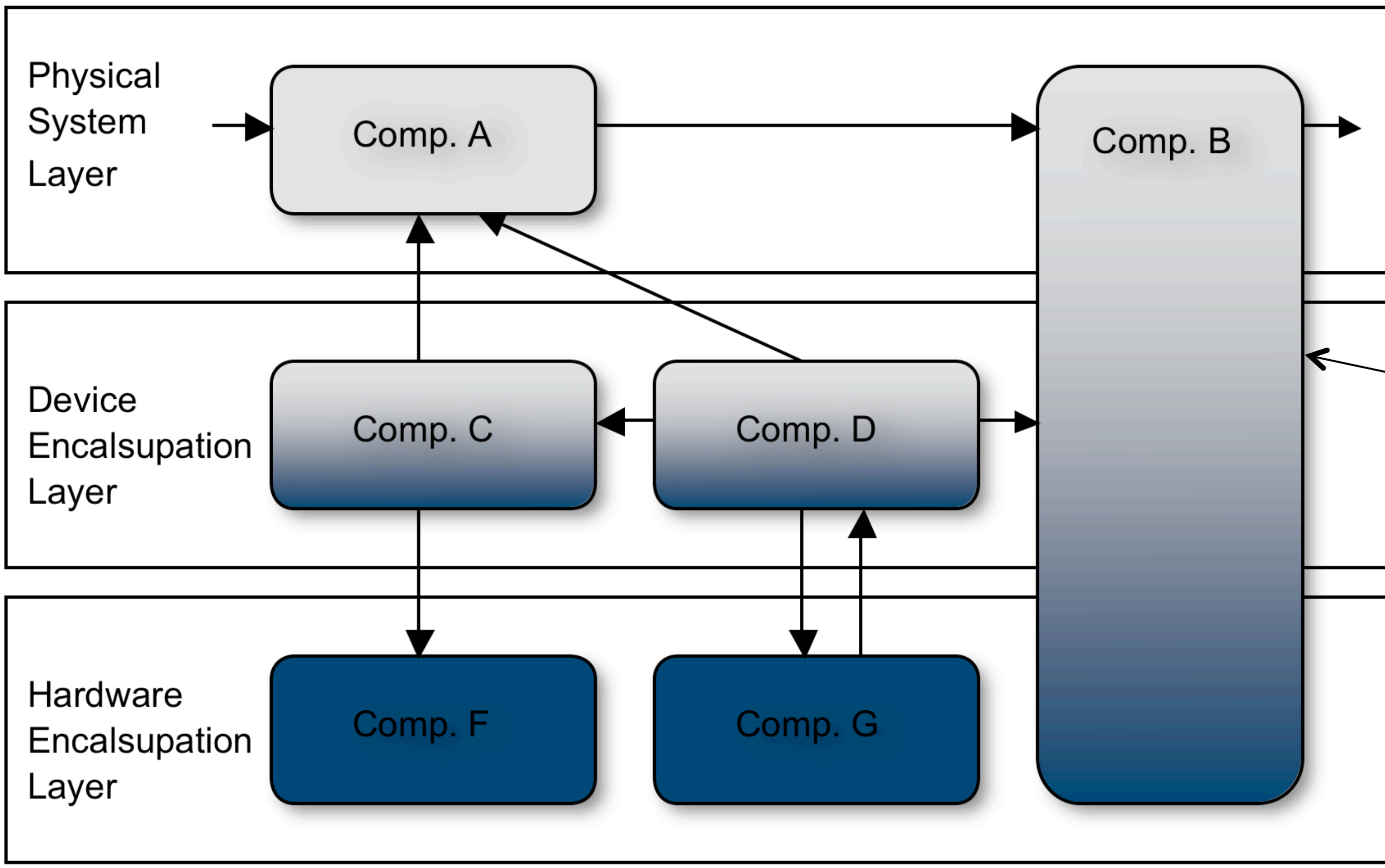
Gain: Easier to correctly meet global requirements, e.g., complex error handling

Loss: More difficult to evolve individual components with new or different error conditions, messages

Monolithic Layer-Spanning Components



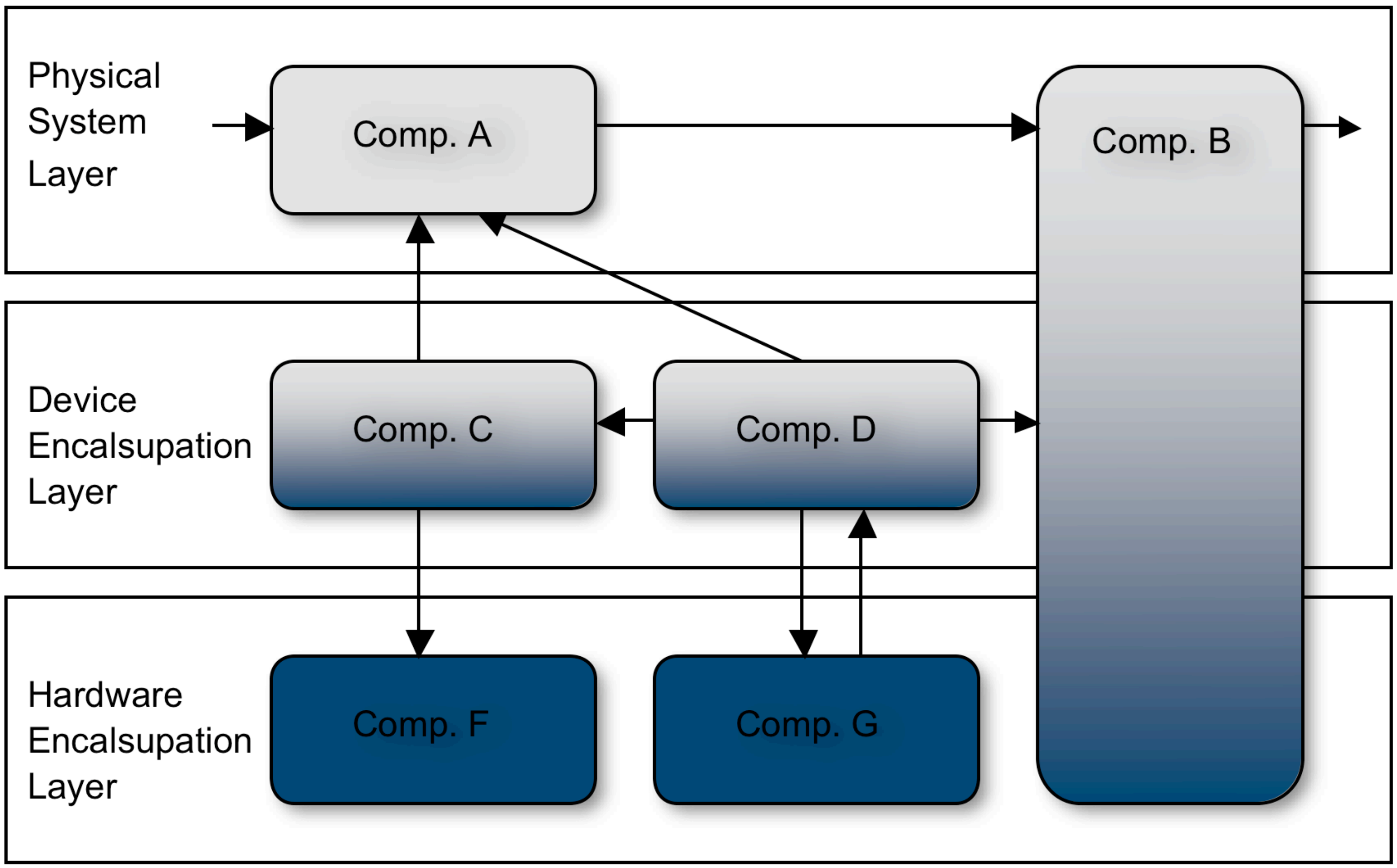
Monolithic Layer-Spanning Components



Most functionality is implemented in layers

Exception granted for functionality with highly complex interactions across layers, e.g., sensors actuators, and computation for automatic parking

Monolithic Layer-Spanning Components



Gain: ability to coordinate work within component B; use co-located team

Loss: potentially very difficult integration with other components, unexpected interactions

Carnegie Mellon

An Observation

- Centralized versus decentralized decision-making
  - Centralized can globally optimize decisions in stable environments
  - Centralized is bottleneck in highly dynamic environments
  - Centralized is slower, longer and larger information flows
  - Decentralized may be better for solving immediate problem, may cause future problems
- Fundamental approach: solve the hardest problems by assigning all the closely-related work to a single, co-located team, manage the rest

COASTE

Carnegie Mellon

Some Research Issues

- How to capture the organizational part?
- How to capture the dynamism that drives the style/pattern?
- Dimensions of coordination capacity?
  - Communication bandwidth
  - Tendency to cooperate
  - Correct anticipation
  - Background knowledge

COASTE