An Approach to Implementing Self-Healing Systems

Nathan Combs, Jeff Vagle
BBN Technologies
Cambridge, MA
ncombs@bbn.com, {jvagle@attbi.com}

*This work is sponsored by the Defense Advanced Research Projects Agency (DARPA). In particular we acknowledge the support of Dr. John Salasin and the DASADA (*Dynamic Assembly for System Adaptability, Dependability, and Assurance) *program. Contract Number: F30602-00-C-0203*

Introduction (biases)

- "Infrastructure impose or constrain choice of architecture" *agent-based architecture, distributed blackboard, pub/sub style*

- "Infrastructure…[is] the self-healing architecture" *self-healing self-healing emerges from interaction of agents.*

- "Self-organizing" ->
  - "self-assembling" *(agents, workflow, fully distributed components and infrastructure),*
  - "self-healing" *(local constraints, global hints)*

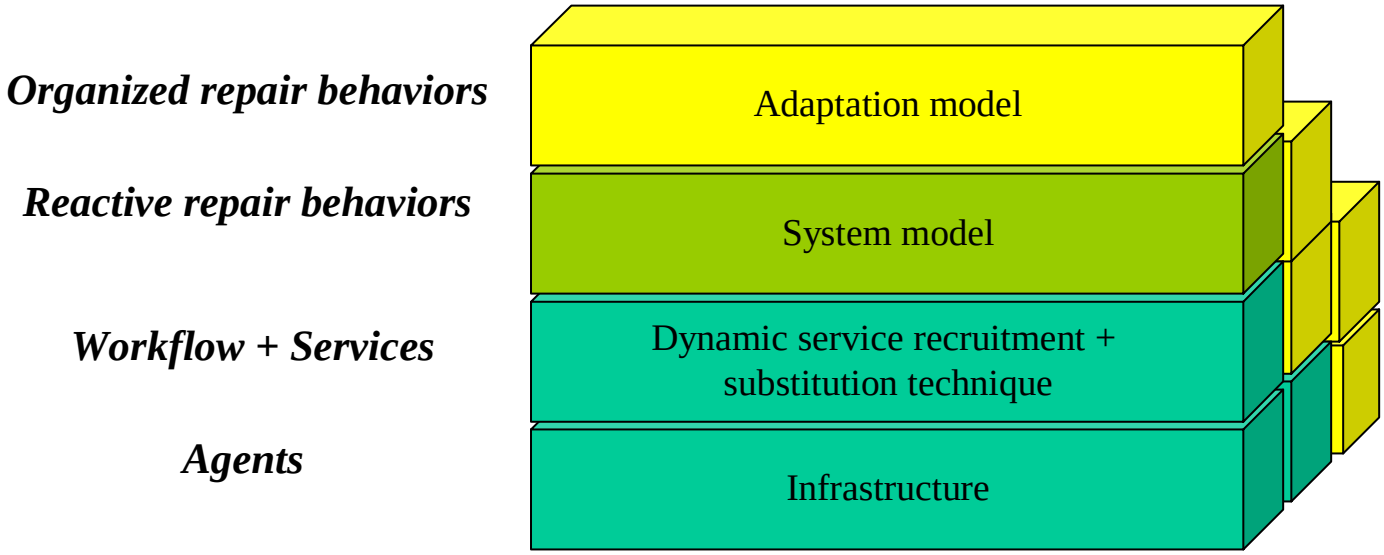
Self-Healing

- Target: legacy loosely-coupled systems.
- Approach: “mirror” vulnerable connectors using an adaptive alternative
  - Use alternative services
  - Use alternative pathways
- Testbed: Adaptive Connector*
  - Agent-based
  - Reactive using gauge**/ sensor inputs

* “Connector” used in the EAI sense (e.g. J2EE Connector Architecture). Alternatively - “Adapter”.
** DASADA

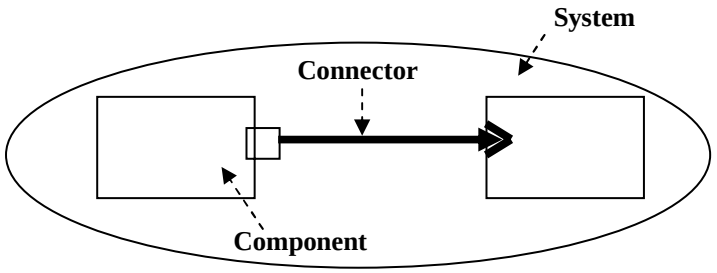
Testbed Objective

- Examine self-healing as an organizing system behavior
- Commit to a testable implementation (DASADA)
  - Complete (albeit thin) slice
  - Based on existing agent-based framework (http://www.cougaar.org)

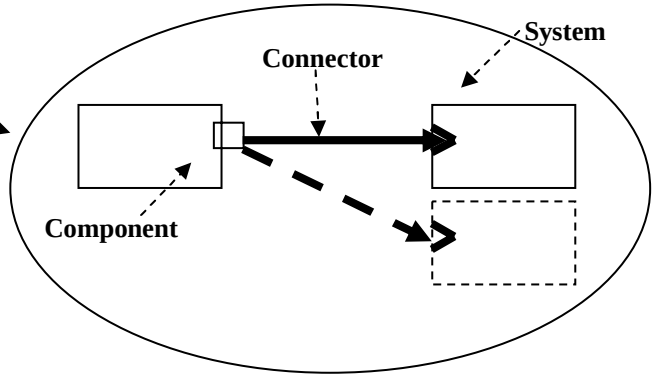


Case Example: Smart Adapter

- Testbeds: 01/02 DASADA demonstrations
- Passes (and transforms) data from Service1 to Service2
- *Source Adapter*. Called directly by source. May invoke target directly or may just pass event.

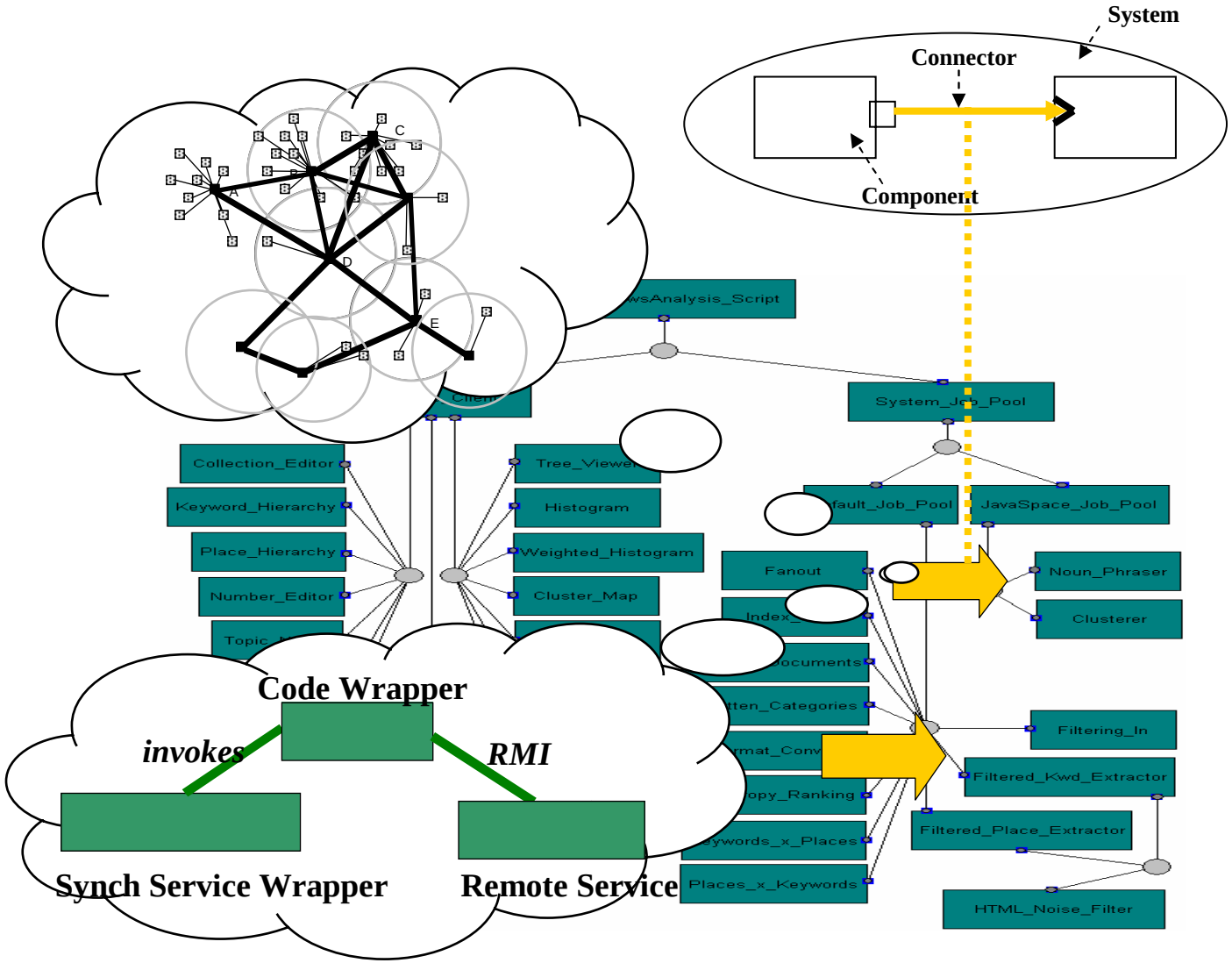


- Adapter model abstracts type instances (substitutes).



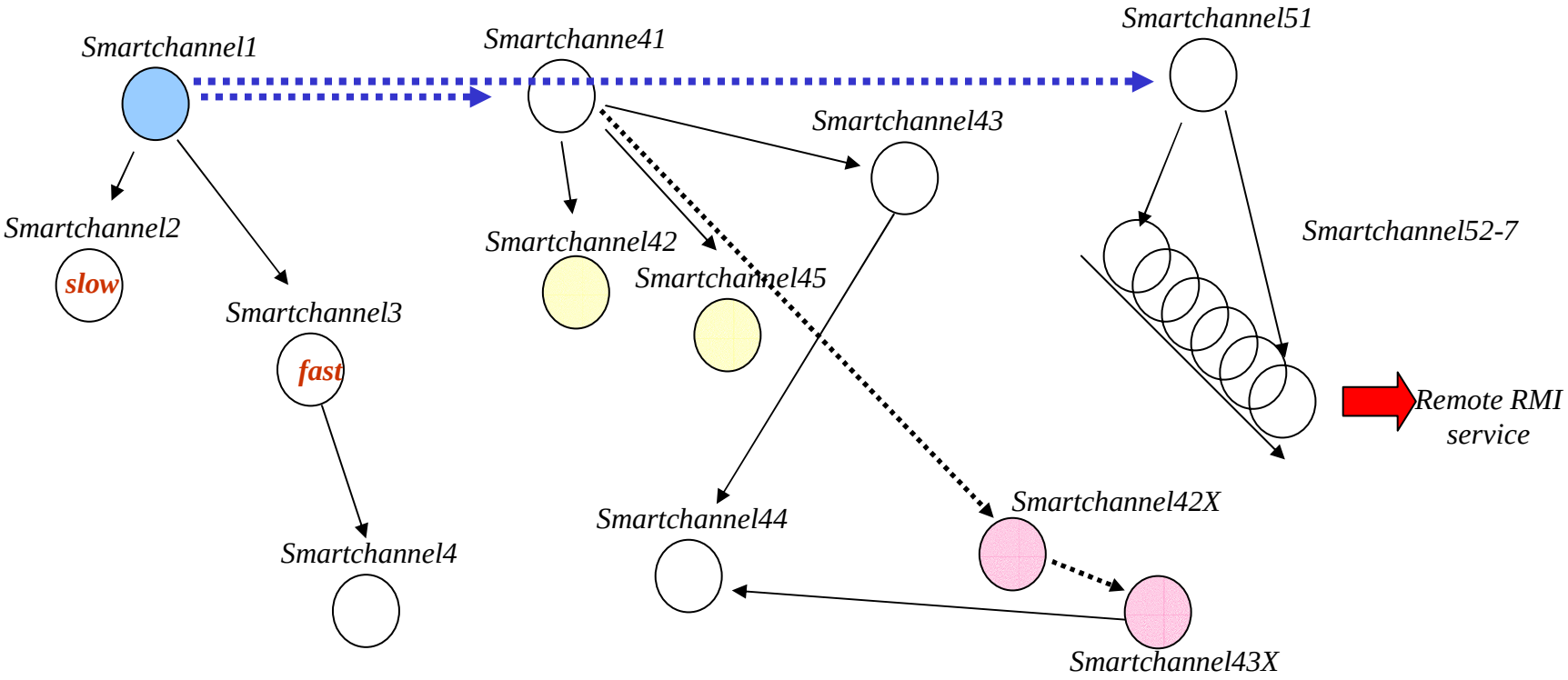
Smart Adapter

- Adapter model abstracts internal details.



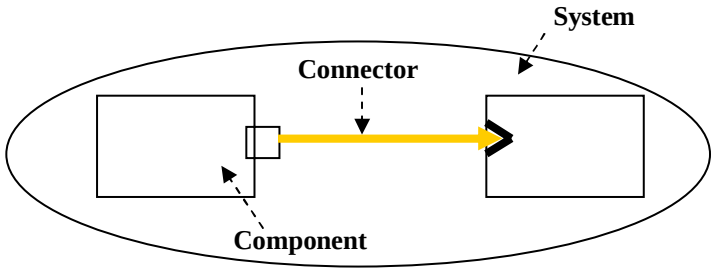
2002 Demonstration System

3 scenario threads, ~18 nodes, ~50 domain components, ~100 infrastructure Plugins



GW- Keyword Extractor scenario *Simulation scenario* *GW- Noun Phraser scenario*

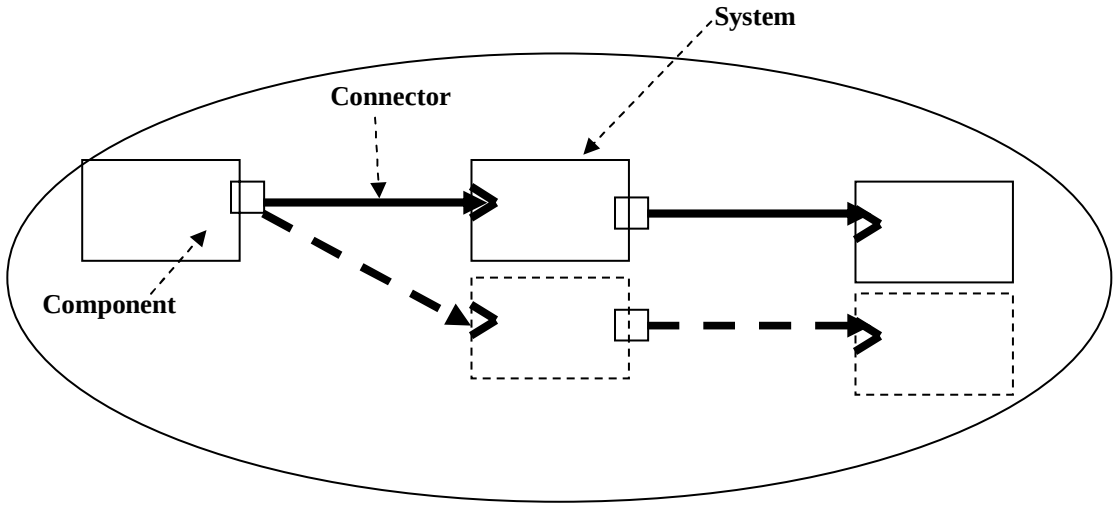
Smart Adapter



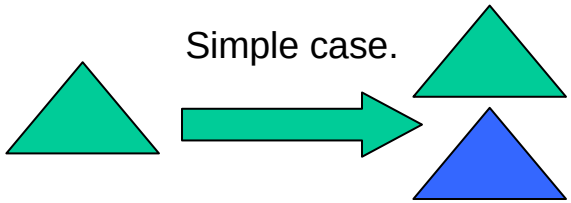
Adapter is implemented using agents.

- Agents connect application components
- Agents can be distributed
- A single agent society may implement multiple Adapters
- Agents are distributed and services are loosely coupled.
- Agents collaborate to implement a large-scale (albeit simple) distributed workflow engine.

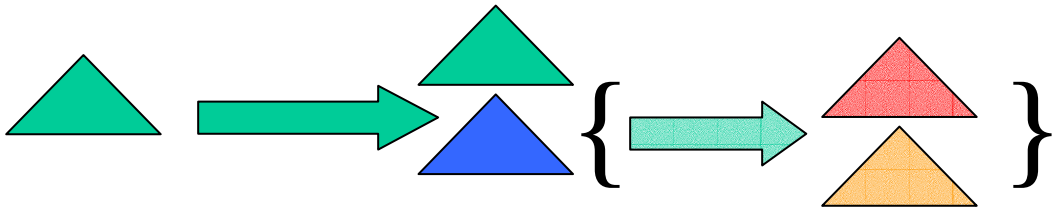
Enables Dependent Flows



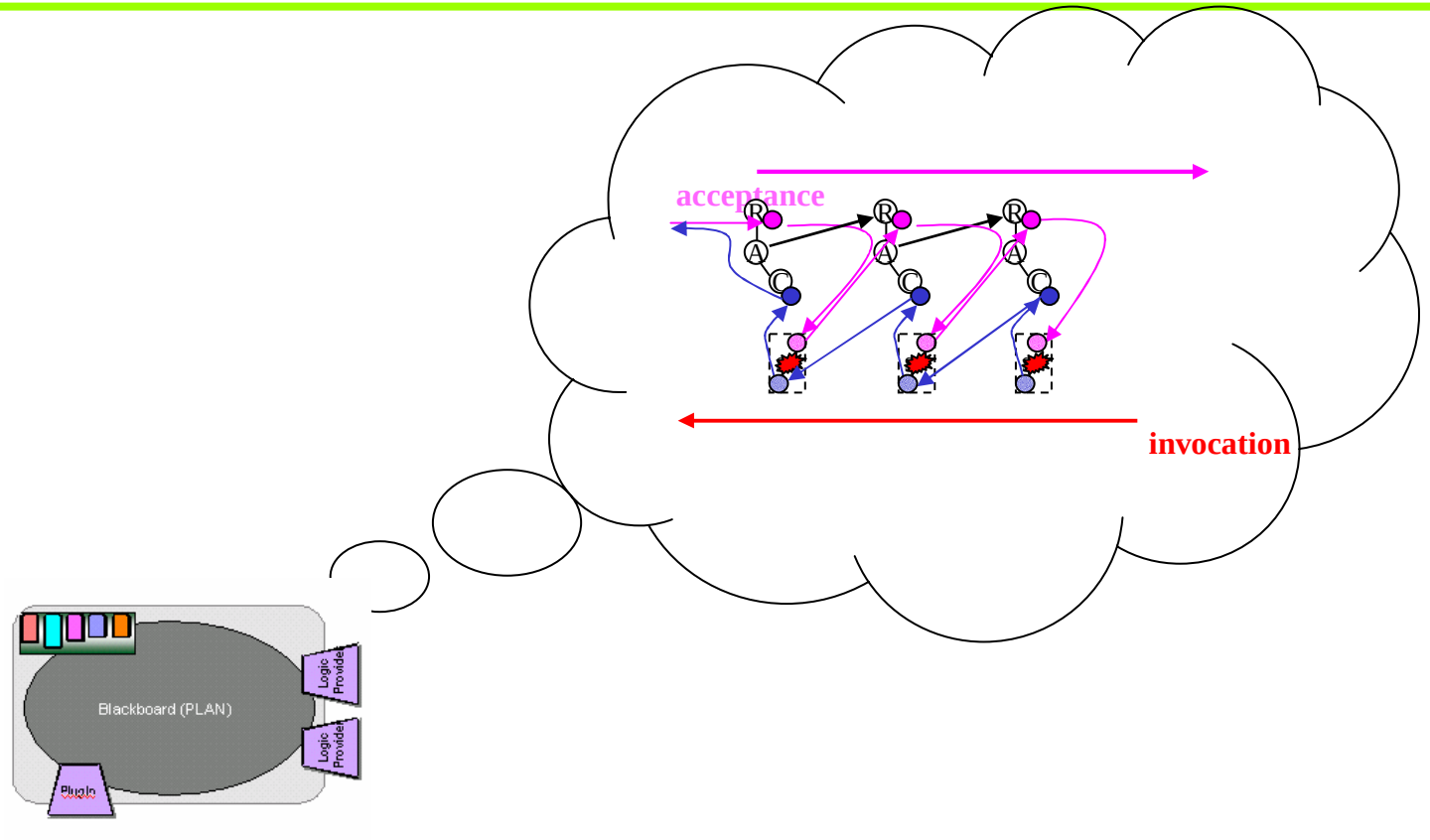
Simple case.



More interesting case.

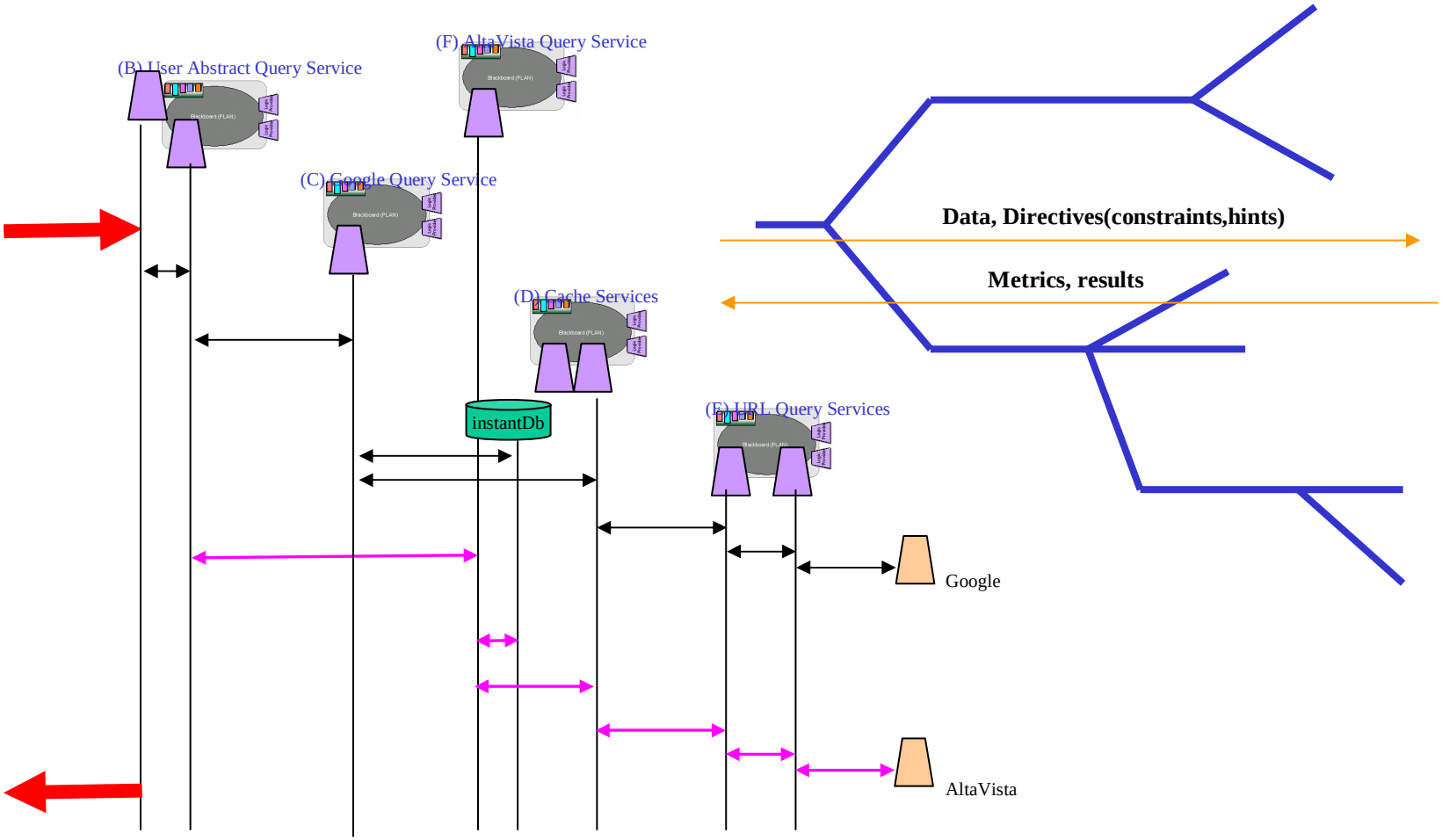


Each Agent is an Island



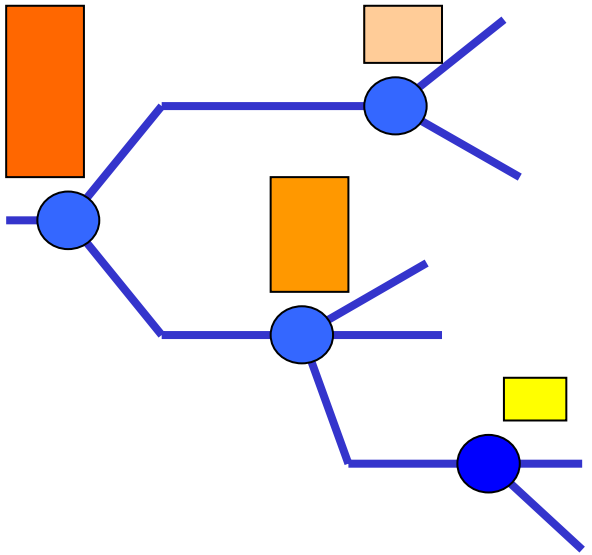
1. Task (request for service) enters
2. Task is decomposed (dependencies)
3. Services are matched to Tasks (contracted)
4. Unmatched Tasks (dependencies) get routed elsewhere

Agent Society Establishes Information Flows



Information Flows Exploited

- Testbed: 02 DASADA demonstration
- Information Flows drive systemic adaptation
  - Unsupervised learning analogy
  - "Over and beyond" local mechanisms (local service substitution, constraint checking)
  - Metrics as they flow-back are examined by agents (acting independently)
  - Metrics used to create Hints (+ Constraints) for future workflows.



Assertions

- *Self-healing self-healing emerges is necessary for fully distributed systems (scalability).*
- *Infrastructure (e.g. agents, distributed blackboard + pub/sub) constraints choices/ approaches.*
- *What can be generalized?*

Backups

Single Agent System; Multiple Adapter Instances