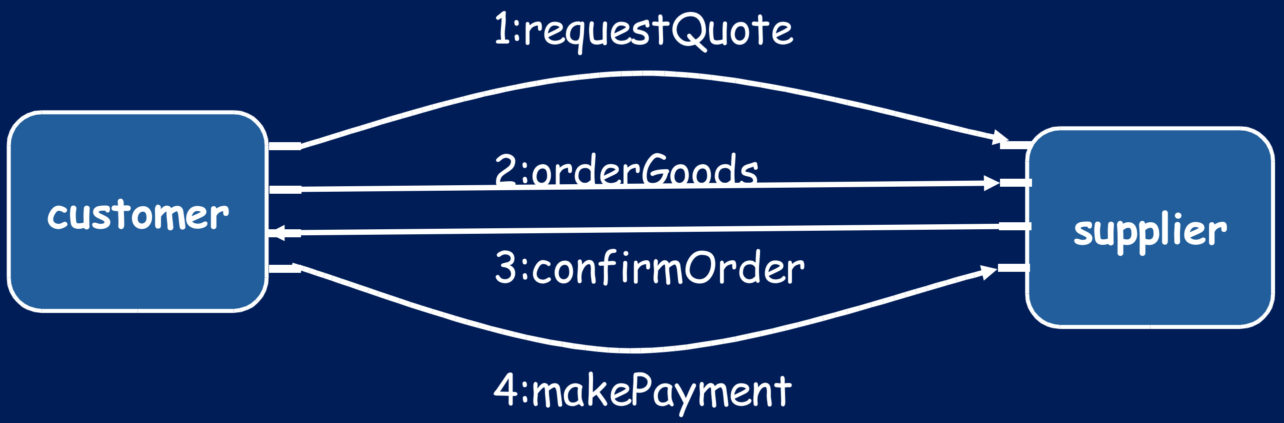
Coordination and composition

Panel on Service Modeling

Fabio Casati

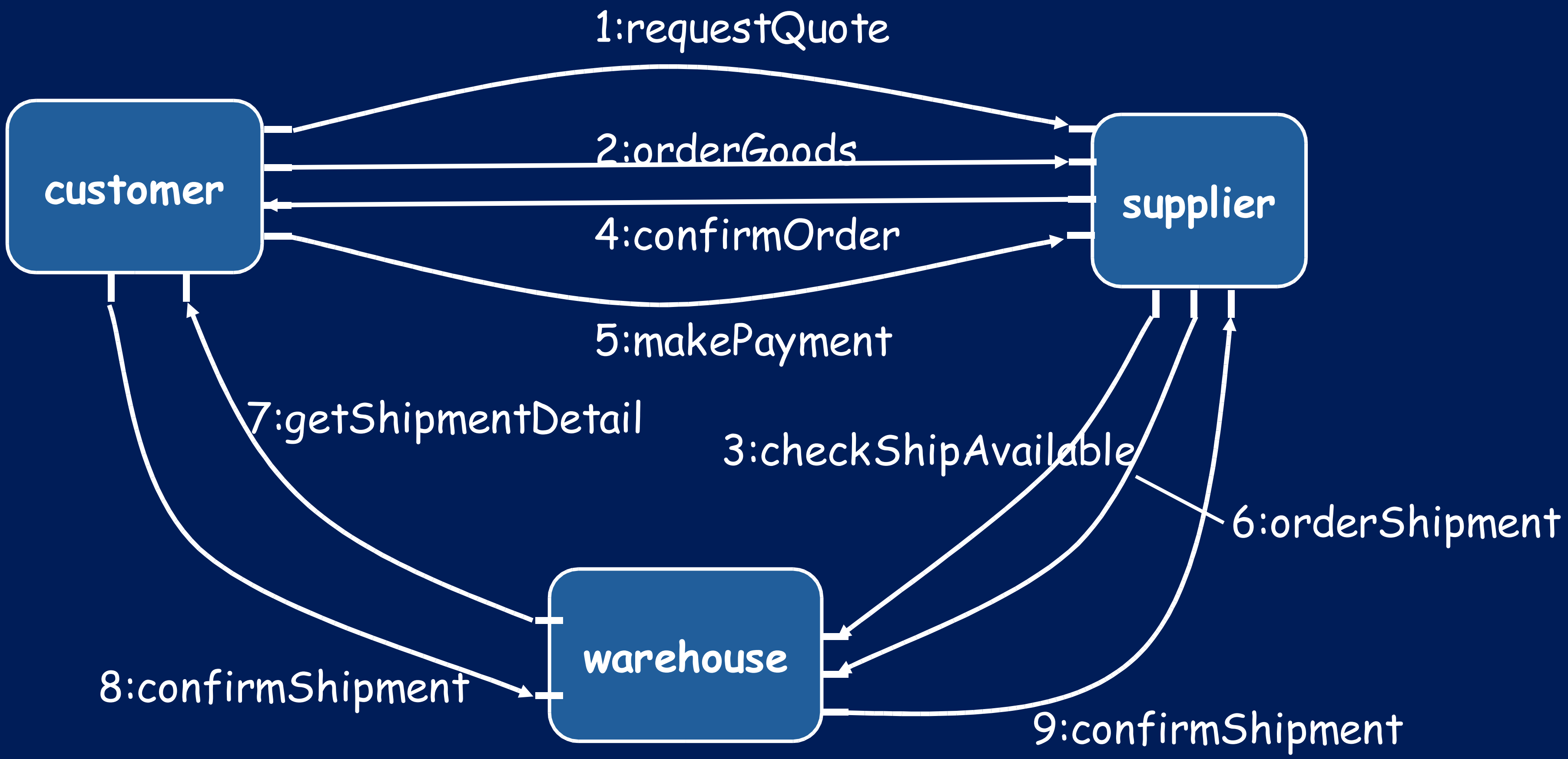
SOC’03 - December 16, 2003

Conversations and protocols



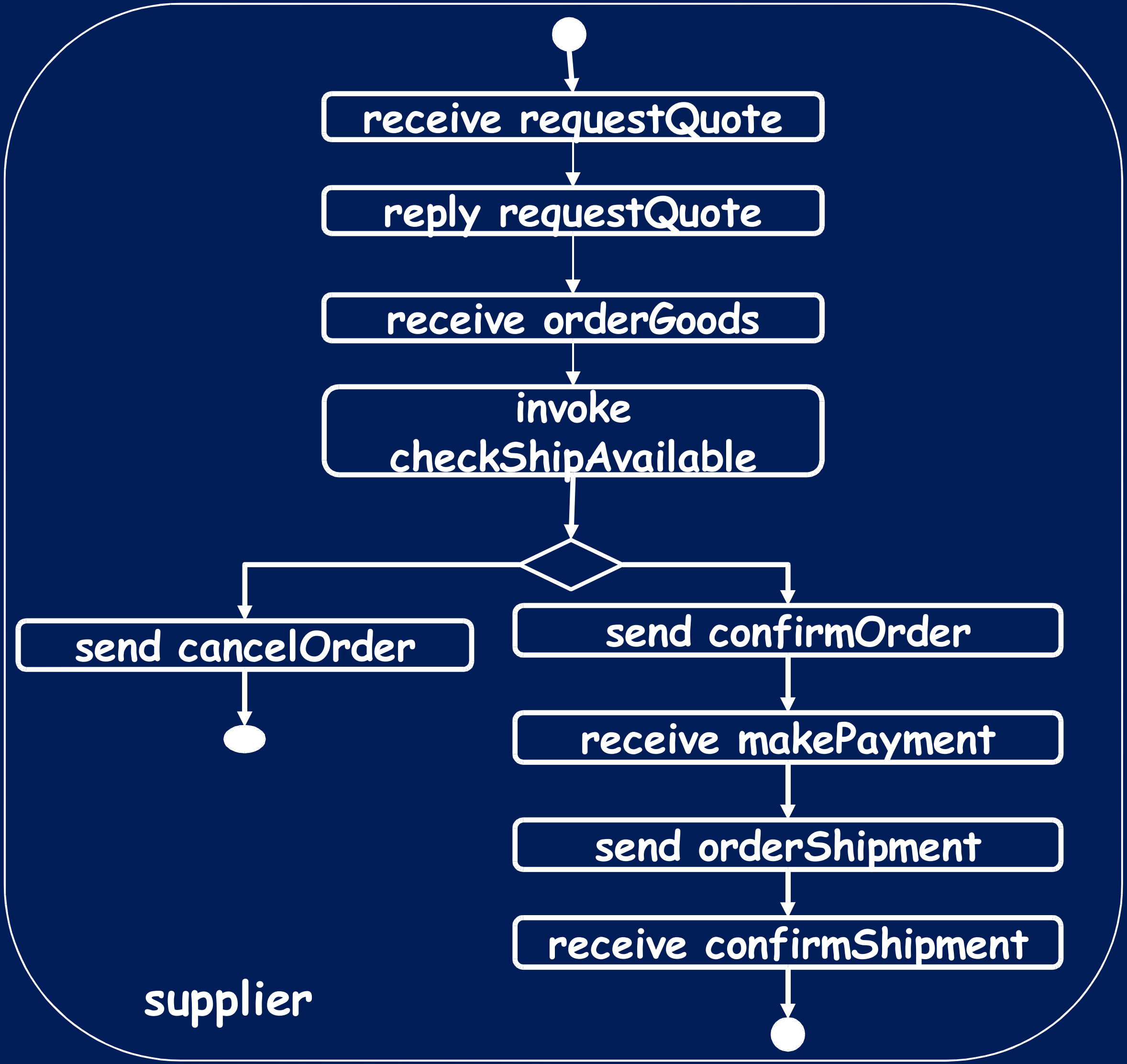
- More than interfaces
- As important as IDLs
- Mindshare, and a real need
    - needed because of loose coupling
    - simplifies development/mgmt

Multi-party conversations



- Tons of different models available

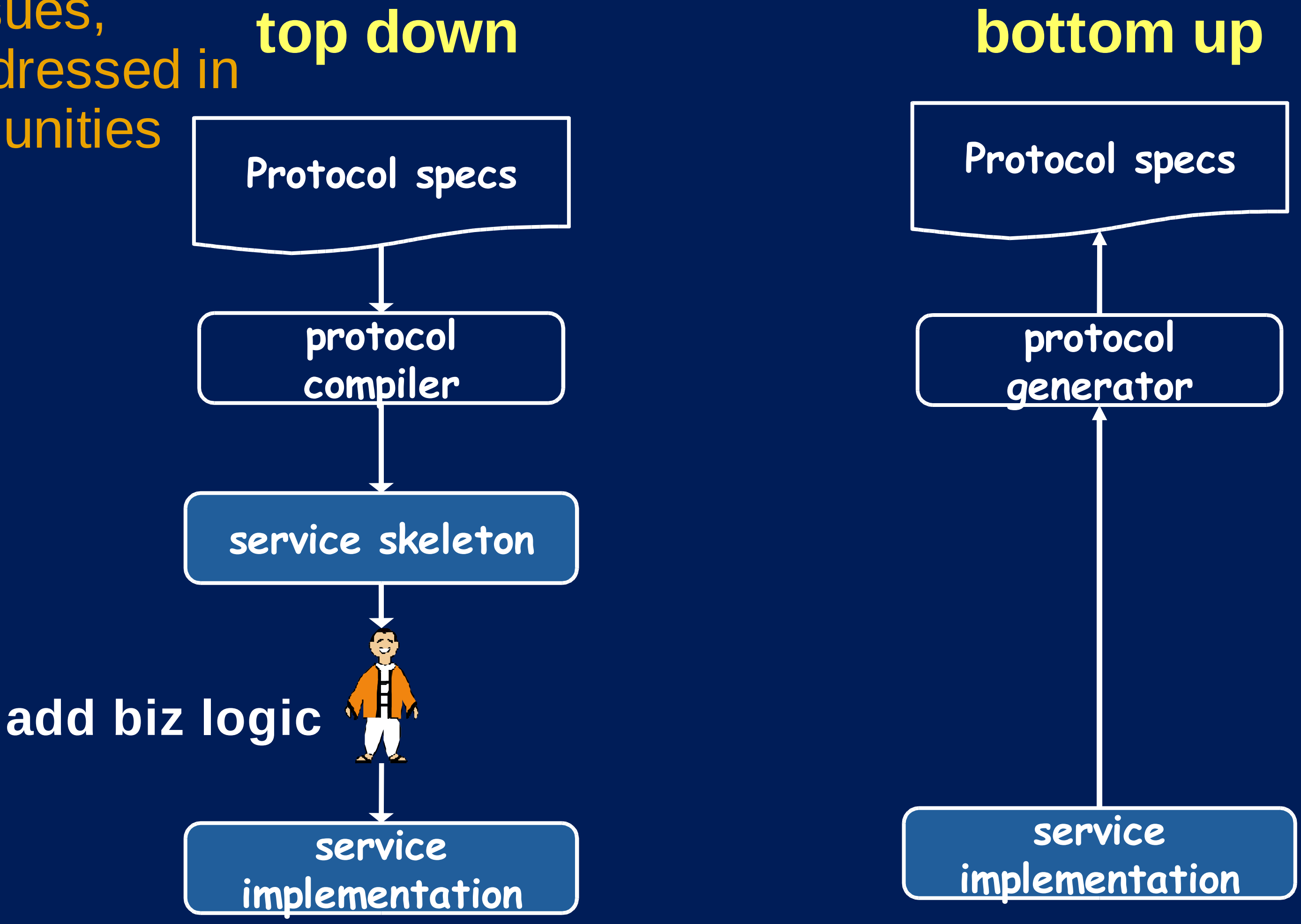
Service-centric conversation models



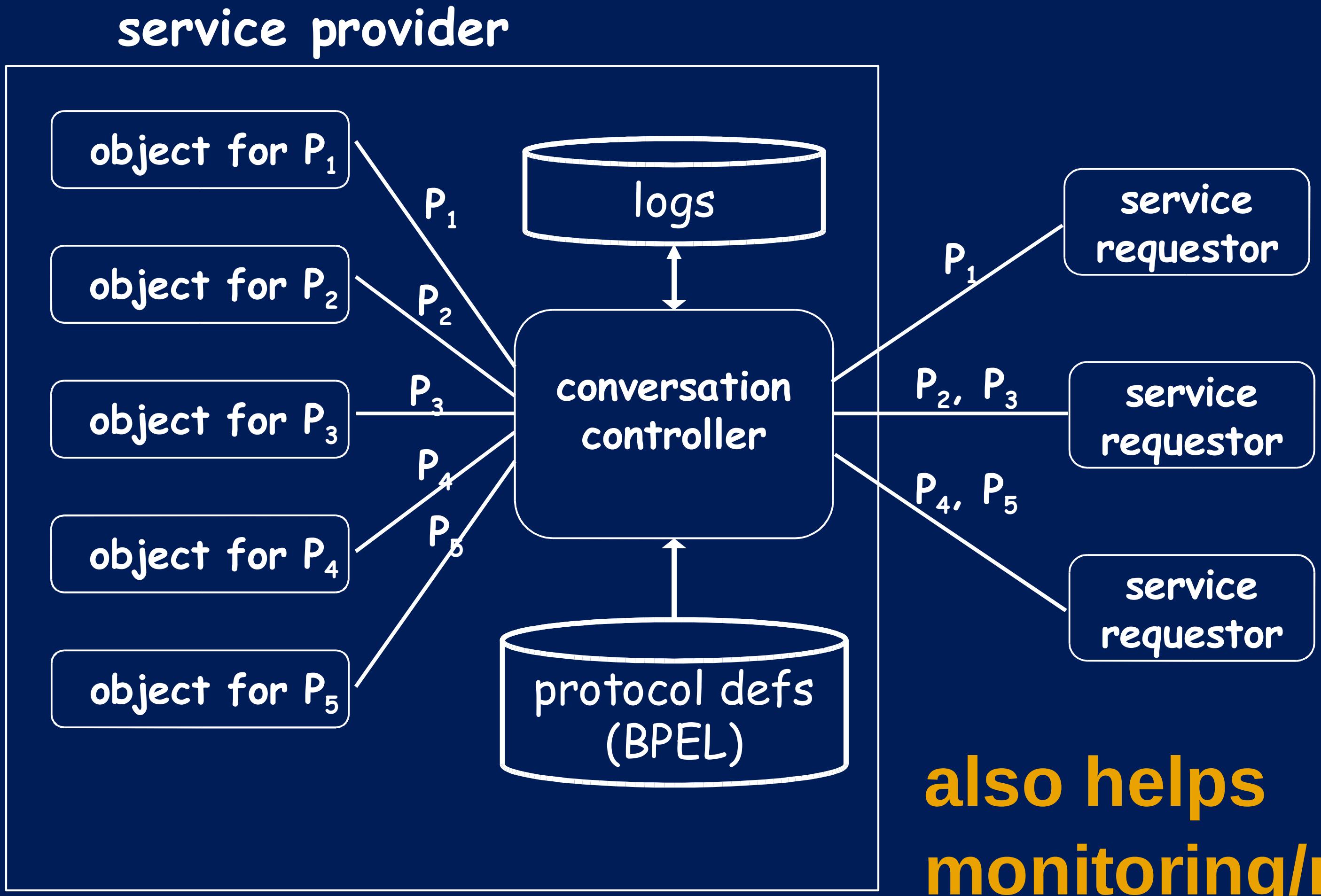
It looks like a “traditional” workflow, but it’s not.

Why protocols? development support

- generate skeletons
- research issues, probably addressed in many communities

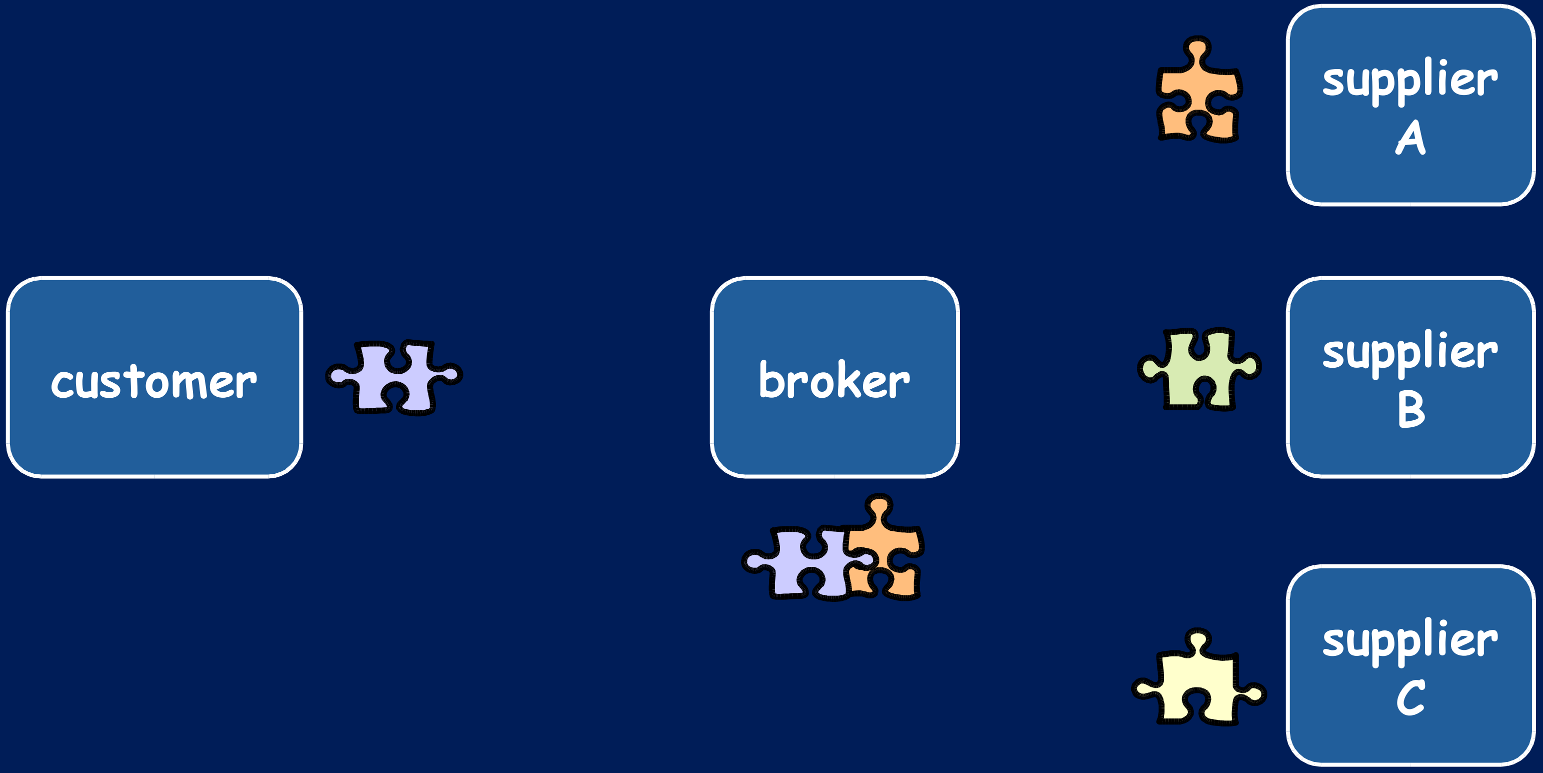


Why protocols? middleware: routing, validation, logging



also helps monitoring/mgmt

Why protocols? matchmaking



- lots of interesting problems wrt **syntactic** compatibility
- in principle, it comes for free! All done by the middleware

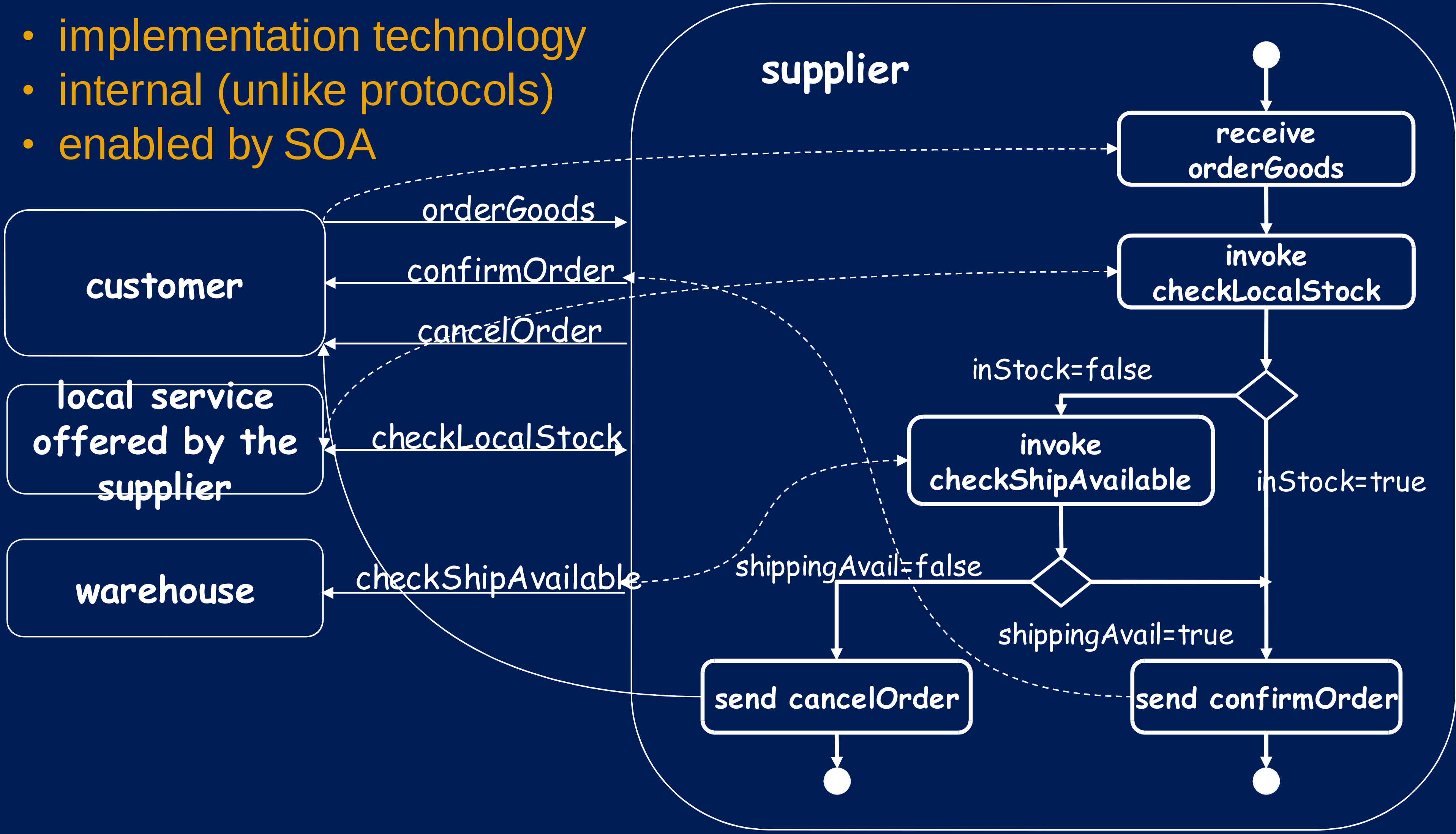
Coordination at different levels

- Transaction
- Security
- Meta-coordination
- …

- Some needed because no central middleware
- interactions among the different protocols (and related middleware) still to be studied

Composition

- implementation technology
- internal (unlike protocols)
- enabled by SOA



What’s new

- It works! (maybe)
  - secret is in the components, not the composer
- Implements a protocol, not an operation
  - interactive/cooperative, not a dictatorship
  - languages designed with this philosophy from the start
  - best aspect of BPEL
- Push model
- Standards (??) and integration with other standards
- Tools: built on top of the middleware stack – much easier to develop and deploy
  - cheaper, even free
  - can install in hours, not weeks

why XML?

- people understand it
- tools understand it
  - validate, parse, query,…
- consistent with other WS standards

- end users do not care

Is the flow representation enough?

- key is standardization of components and better tools
  - browse services
  - drag ‘n drop services into the canvas
  - integrate composition with ad hoc programming
  - testing, tracking, analysis
- this is more important than the flow model
- issues such as brokering, dynamic binding, semantics, automated intelligent composition, not essential now.

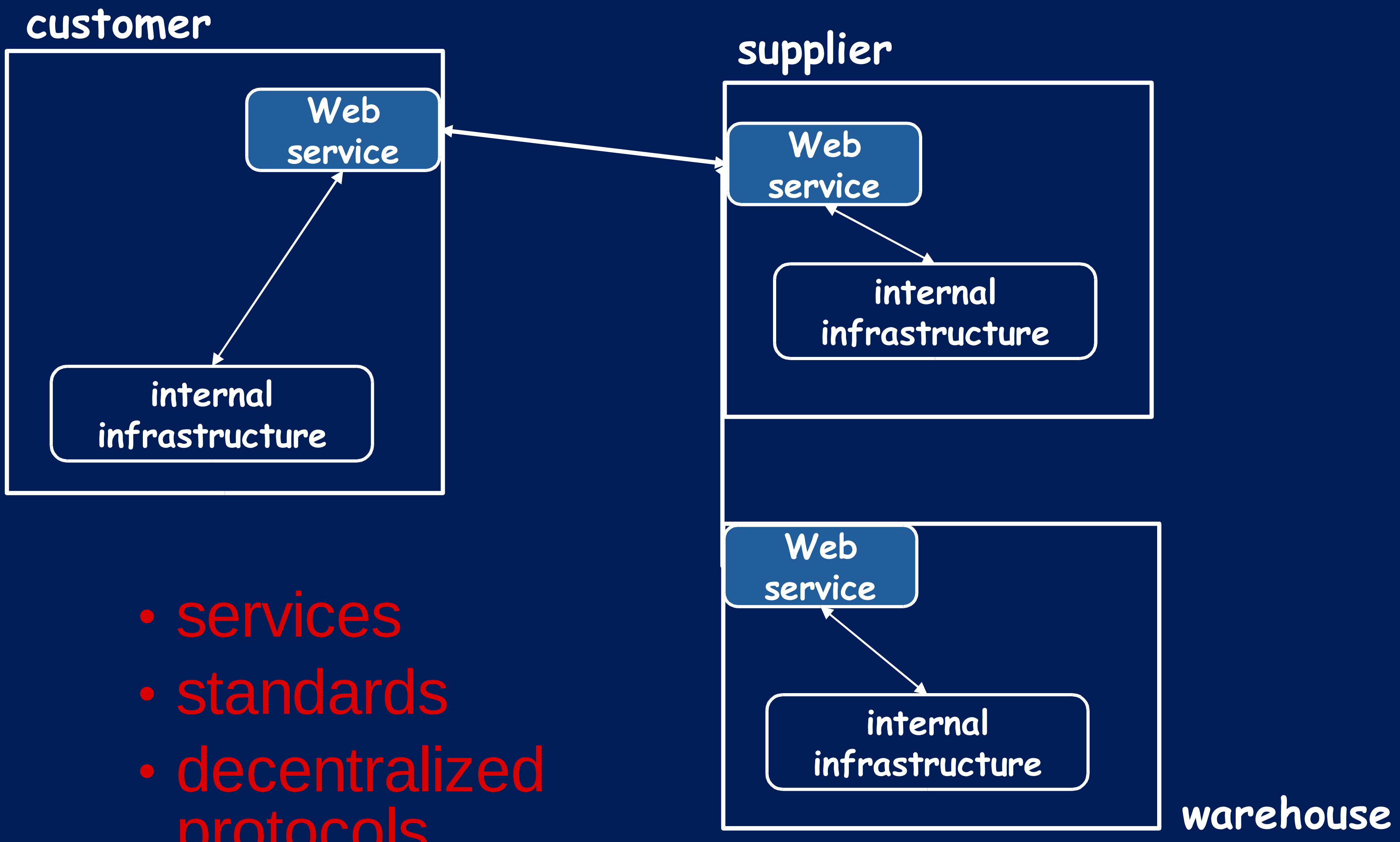
hp®

invent

Essentials

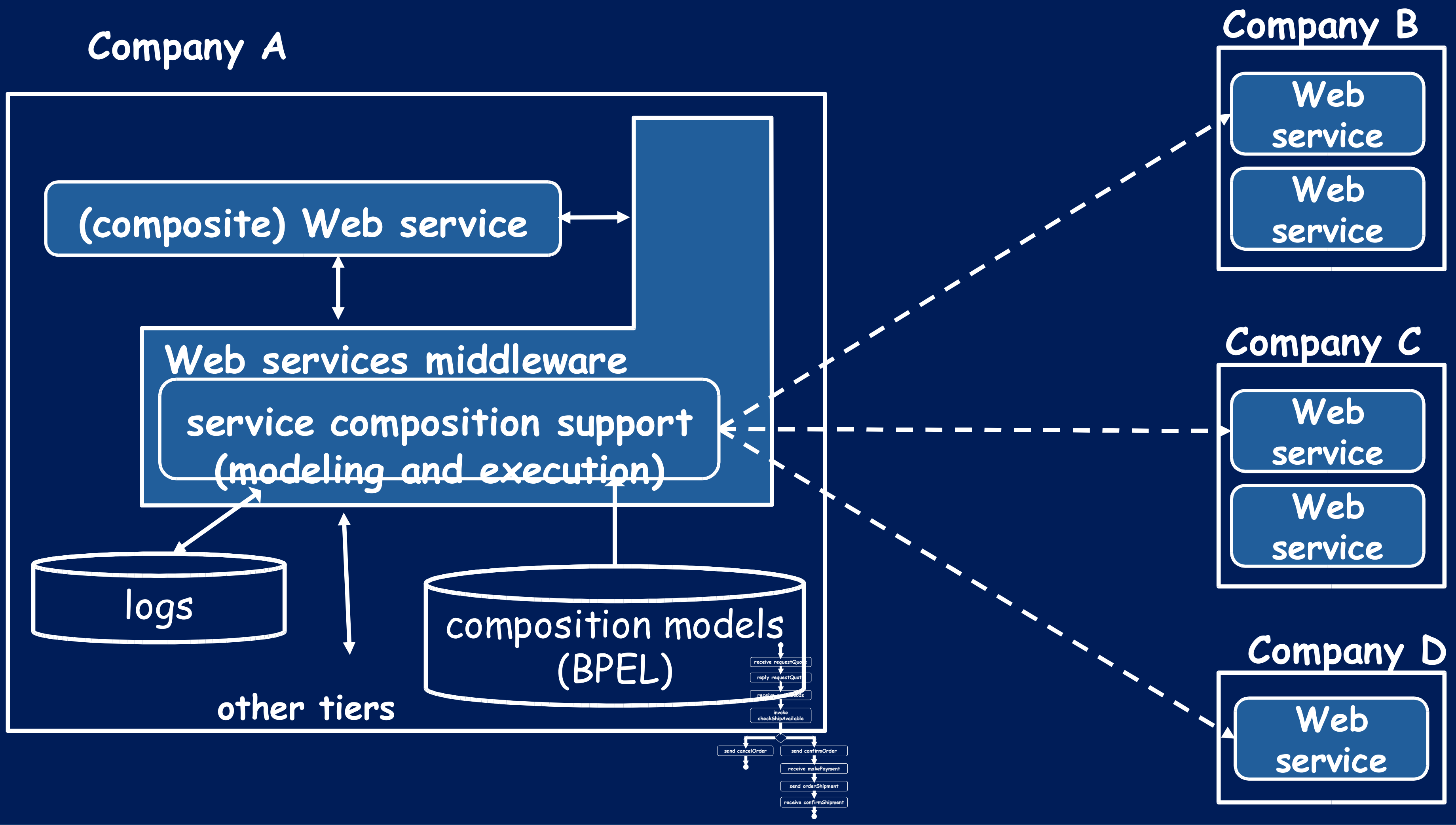
- composition great opportunity
  - can succeed where previous attempts have failed
  - why?
- Implement a service, not an operation
- External and internal specifications
  - coordination and composition go together
- Standardization (?)

Essence of Web services approach



- services
- standards
- decentralized protocols

Composition middleware



Open problems

- Metric definition and computation framework
  - generic but simple
  - SLAs and contracts
- How to bundle intelligence into the tool
  - correlation, prediction, intelligent analysis, sensitivity
  - outliers
  - specialized for some problems, or generic
- Feedback
  - how to provide *controlled* automation
  - manageable processes
  - manageable services
  - framework to *easily* define and manage policies

Message

- Management as a key problem/opportunity in WS
- Web services enable biz. aware management
  - B2B, service oriented architectures
  - Standards
- Conversations, compositions, correlation
- new problems, higher expectations
  - **data mining is a key technology**
- Assess, advice, act

Grid computing

- "…a way of organizing computing resources so that they can be flexibly and dynamically allocated and accessed, often to solve problems requiring many organizations' resources…" [OGSI Primer]
- originally, a network (protocols and conventions) for sharing cycles for compute-intense scientific applications
- now, a service-oriented connecting architecture for collaborative applications requiring access to global resources

Global Grid Forum (GGF)

- GGF is the standards body for the Grid
  - GGF is to Grid as W3C is to the Web
  - Composed largely of academics, but being increasingly influenced by industry (IBM, Sun, Fujitsu, HP, Platform, Avaki, …)
- Community has resource sharing as a mindset
  - Many come from scientific background where resources are scarce, and sharing is common
  - Growing up to broader view of “resources” and stricter need for access control

OGSI and OGSA

- OGSI (Open Grid Services Infrastructure):
  - A Service Component model building on Web Services/WSDL
    - Support for: transient services, Life-cycle, Registration, Notification, etc.
- OGSA (Open Grid Services Architecture)
  - An umbrella for identifying services of importance to the Grid
  - All services will be OGSI-compatible
  - Examples: Logging, Workflow, Reservation, Instrumentation/Monitoring, Cycle scavenging, …

Grid services

- All services in the service-oriented architecture of the Grid must adhere to a service-component model prescribed by OGSI.
- In particular, OGSI has defined
  - extensions to WSDL 1.1 to encapsulate state of a service
  - a port type called “GridService” that provides basic functionality such as identification and lifecycle to every service
  - several port types (factories, service grouping, agreements, etc) for sharing resources represented as services