

SOA Distilled

Udi Dahan – The Software Simplist

.NET Development Expert & SOA Specialist
Microsoft Solutions Architect MVP

Udi Dahan - The Software Simplist

Consulting, Mentoring & Training
Helping you Keep It Simple

www.UdiDahan.com

email@UdiDahan.com

**SOA is all about
reducing coupling**



What is coupling?

- A measure of dependencies
- If X depends on Y, there is coupling between them
- 2 kinds of coupling: Afferent (Ca), Efferent (Ce)

What is coupling?

- Afferent coupling (C_a) – who depends on you
- Efferent coupling (C_e) – on who you depend

What is coupling?

- If X depends on Y then:
- X is efferently coupled to Y
- Y is afferently coupled to X

Coupling at the systems level

- C_e : The number of classes in system X that depend on something belonging to system Y
- C_a : The number of classes in system Y that something in an external systems depends on

Coupling at the systems level

- Ce: The number of classes in system X that depend on something belonging to system Y
 - If Y changes in some way, how many changes do we have to make in X?
- Ca: The number of classes in system Y that something in an external systems depends on
 - In what ways can we change Y safely?

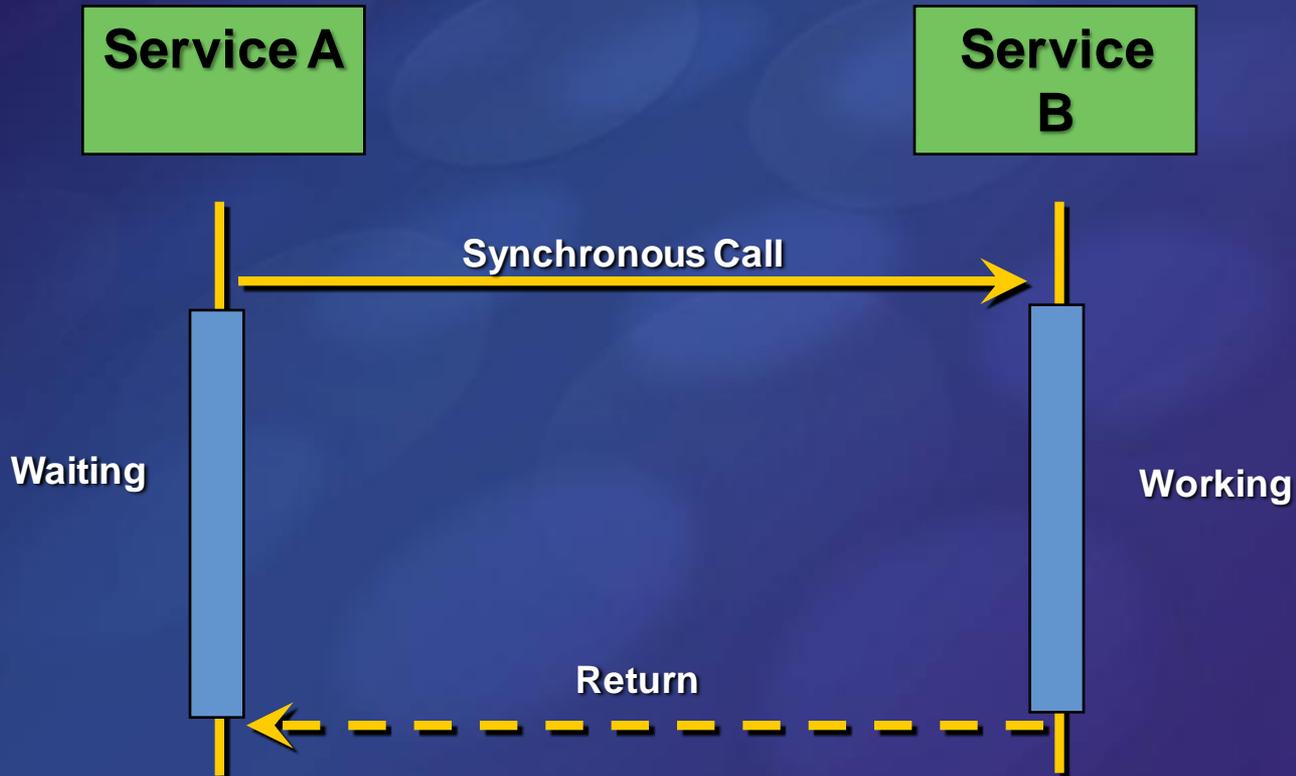
Loose Coupling at the systems level

- Attempt to minimize afferent and efferent coupling
- Zero coupling isn't really possible
- 3 Different aspects of coupling for systems:
 - Platform
 - Temporal
 - Spatial

Coupling Aspect #1: Platform

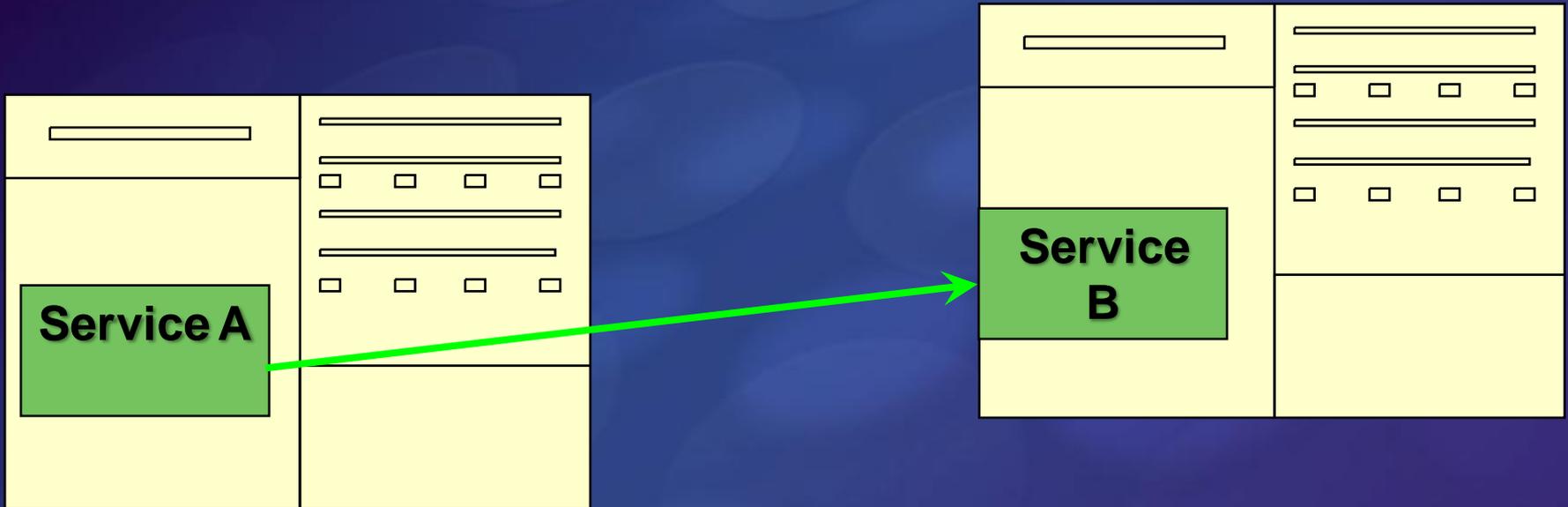
- Also known as “Interoperability”
- Using protocols only available on one platform
 - Remoting
 - Enterprise Services
 - Datasets over Web Services
- One of the famous 4 Tenets:
 - “Share contract and schema, not class or type”

Coupling Aspect #2: Temporal

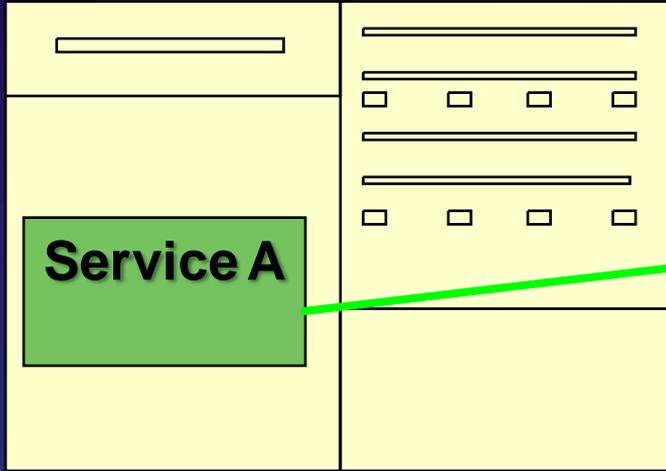


Processing time of Service B affects that of A

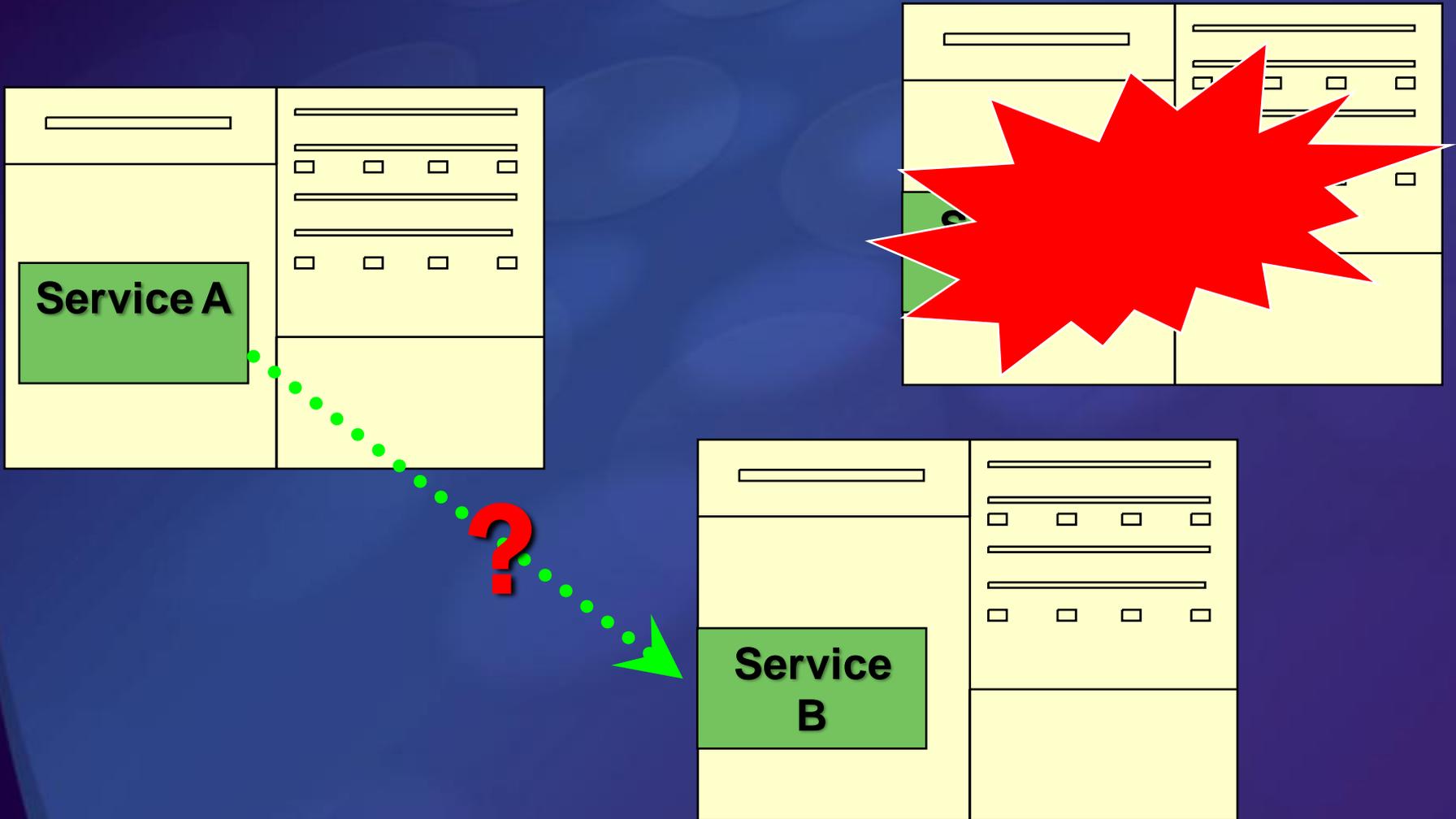
Coupling Aspect #3: Spatial



Coupling Aspect #3: Spatial



Coupling Aspect #3: Spatial



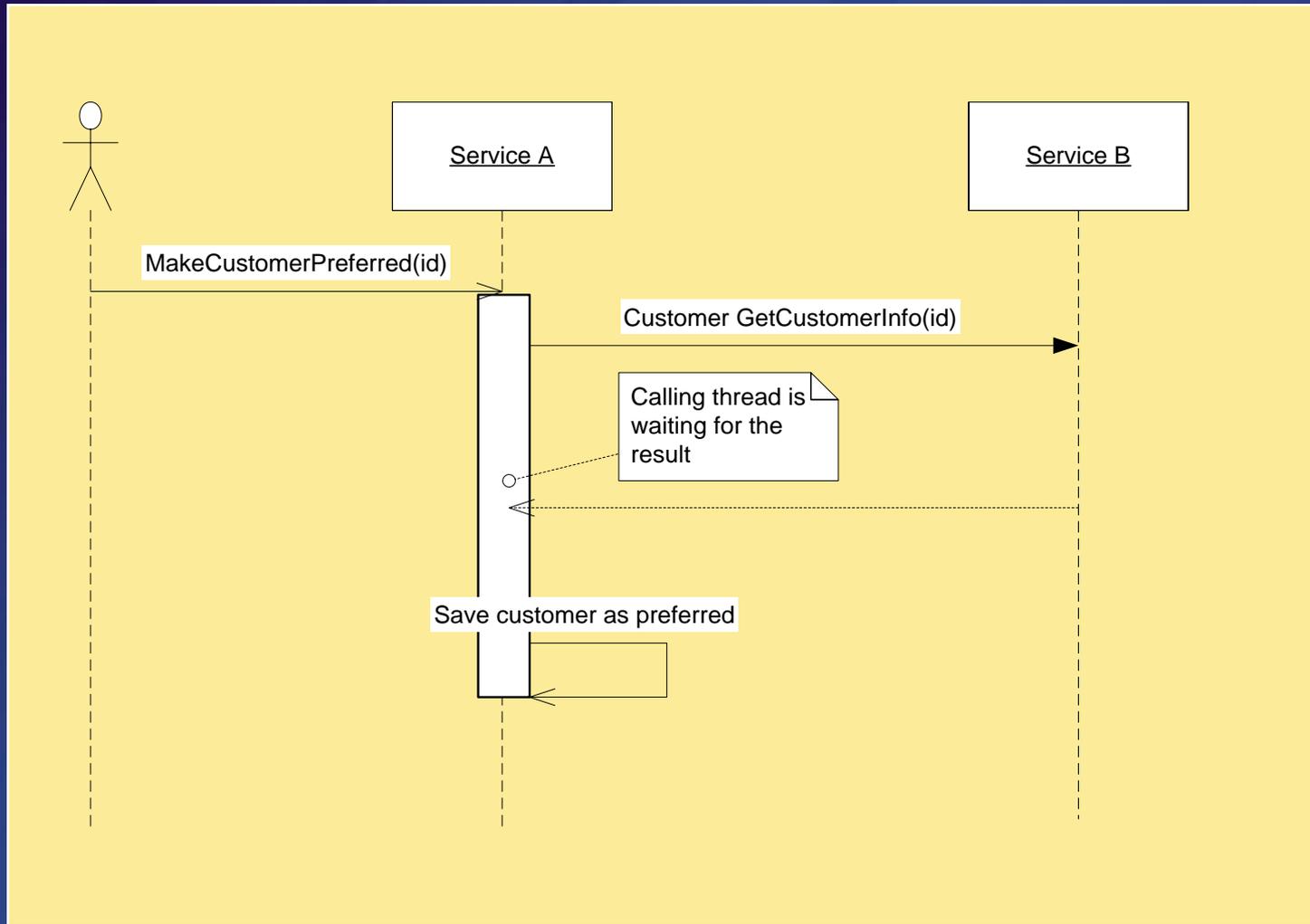
Can communication automatically continue?

Coupling Aspects: Solutions

Coupling Aspect #1: Platform

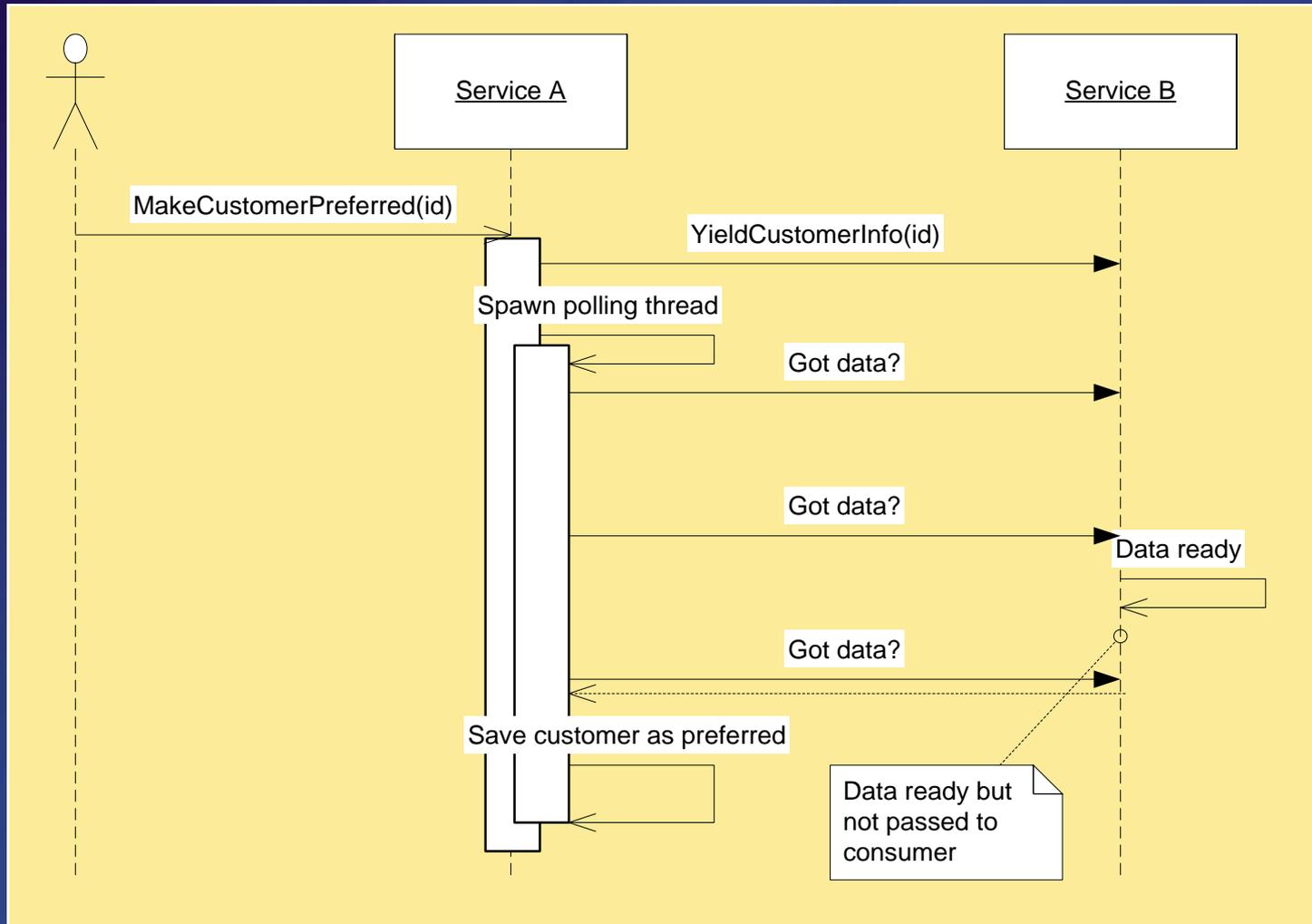
- XML on the wire.
- XSD (schema) describing XML structure
- Use standards based transfer protocol like http
- Standards based description of message flow
 - WSDL (only supports request/response)
 - SSDL (supports richer semantics)

Coupling Aspect #2: Temporal - 1



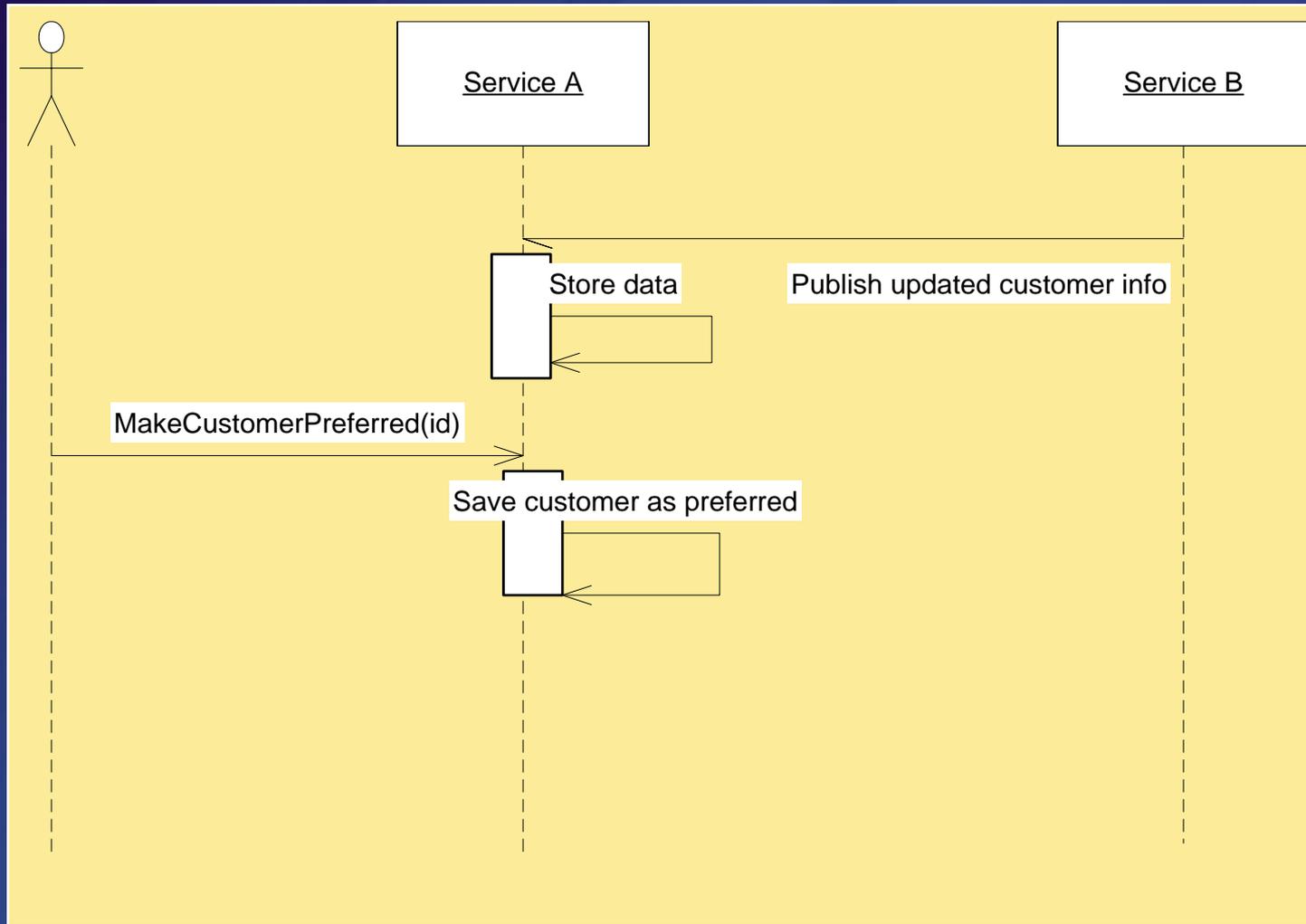
Bad. Resources are held while waiting.

Coupling Aspect #2: Temporal - 2



Resources are held while waiting. Increased load on service B per consumer (impacted by polling interval)

Coupling Aspect #2: Temporal - final



Good. By separating (in time) the inter-service communication and the request handling

Transactions & Temporal Coupling

- Asynchronous messaging makes it difficult to have transactions cross boundaries
- Don't let transactions flow between services
- Autonomy means not letting any one lock something that's yours

Coupling Aspect #3: Spatial

- Application level code should not need to know where cooperating services are on the network
- Delegate communications to “something else”, let’s call it an “agent” for now.
 - `myAgent.Send(message);`
- How does the agent know which destination to send the message to?

Coupling Aspect #3: Spatial

- Either the agent uses communications like multicast that don't require knowledge of specific locations.
 - Messaging infrastructure that supports topics does this internally (JMS for example)
- Or the agent needs to communicate with other agents on the network to know where everybody is.

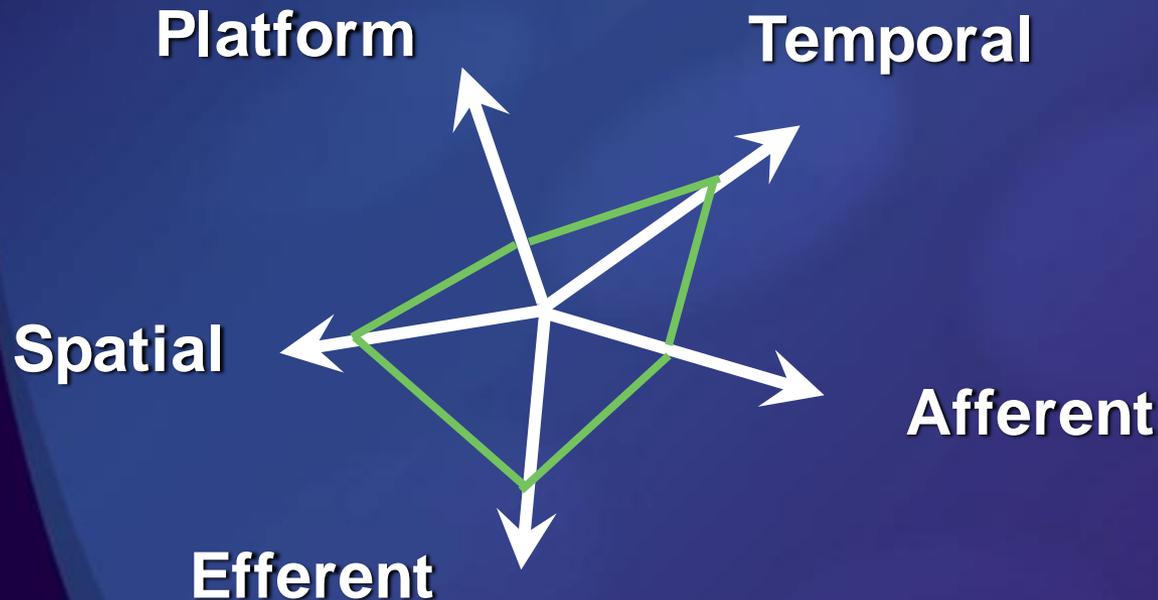
Coupling Aspect #3: Spatial

- But if the application code doesn't tell the agent which *logical* destination to send the message to, how would the agent know?
- If there was a direct mapping from message type to logical destination, then this API would be enough:

```
void Send(IMessage message) ;
```

Summary

- Loose coupling is more than just a slogan
- Coupling is a function of 5 different dimensions



Thank you

Udi Dahan - The Software Simplist

Consulting, Mentoring & Training
Helping you Keep It Simple

www.UdiDahan.com

email@UdiDahan.com