RESTful Conversations

Position Paper by Gregor Hohpe, www.eaipatterns.com

Workshop on Microservices at SATURN 2015

Interactions and Conversations

Message Exchanges between participants can be split into 2 levels:

- Basic interactions that transport data from A to B
- Longer running conversations that include state, error conditions, and often have domain-specific semantics

The dividing line it not 100% sharp, but it's relatively easy to come up with prototypical examples: *Request-Response* or *Fire-and-Forget* belong to the former and *Reaching Consensus* or *lease renewal* to the former category.

Hypothesis 1

Based on this distinction, we can form the following hypothesis:

> *The simpler the basic interaction model, the richer the conversations that take place.*

For example, if basic interactions include transactions and reliable delivery, participants need not worry about such quality-of-service elements. However, if the basic interactions luck such features, conversations such as *Request-Response-Retry* or *Try-Confirm-Cancel* are used by the participants to achieve similar properties.

ReST uses a simple interaction model, hence ReST conversations are an important element of describing the exchange of information between participants.

Hypothesis 2

> *Conversation state can map to participant state or message state.*

Conversations are stateful. This state can be represented inside the participants, equipping messages with a *Conversation Identifier*, or inside the messages, leaving the participants stateless.

ReST favors embedding state inside the message as hypermedia. This eliminates the need for a priori conversation definitions as is the case with Orchestration or Choreography.