There can be only one!
The future of Application Layer Protocols
Henrik Frystyk Nielsen
Purpose of this Talk

• Convergence of Application level Protocols
  – …and the emergence of the Big Hammer Protocol with the small heart
  – Similar to what is happening to XML
  – Working towards a framework
  – Avoiding flag days

• What a Framework Doesn’t Solve
  – Distributed Extensibility vs. interoperability
HTTP/1.1 - The Big Fire Fighter

- Main purpose was to fix three problems
  - Provide a semantically well-defined caching model
  - Support vanity hostnames
  - Limiting waste of TCP connections
- Criteria for solutions was that the user would see a clear win
  - People need personal incentives to change
  - Implementors need clear market benefit to implement
- Goal was Achieved - deployment is underway
  - Performance has improved a lot
Hmm, but how do we Extend it?

- Extensibility wasn’t part of the basic 1.1 package
- HTTP is not a centrally controlled protocol
  - Has maybe never been
  - It’s extended by everybody for any possible purpose
- No structured way of extending HTTP
  - Extend and hope for the best!
- Layering is implicit - looks and feels like hairball
- No explicit type information
  - Using POST as a tunnel mechanism
  - Reducing HTTP to a byte transport
HTTP at the Center of the Typhoon

- Clear tendency in protocol design from...
- Similar protocols with different background (IMAP etc.), to
- Protocols that are largely copying HTTP (RTSP, SIP etc.), to
- Protocols that use HTTP POST (IPP etc.), to
- Protocols that copy pieces for easier integration with HTTP (TIP etc.), to
- Actual HTTP extensions (DAV etc.)
But Why Only One?

- The Convergence is inevitable!
- Application layer protocol development is driven by application programmers
  - They don’t want to have different protocol engines for highly related tasks
- Users don’t want to know the difference
  - Interested in getting my job done
  - Don’t want to get stopped by error messages: I can’t talk to that guy!
- Shared Platform (not necessarily interoperability)
But Why HTTP?

- Has a lot of flexibility for “creative” usage
  - Many interactions are not defined in the HTTP/1.1 spec
  - For example, is end-to-end really end-to-end?
- Often enough to get the work done
  - Few are interested in the beauty of solutions
- But stretching the limits of its extensibility
  - Interactions are becoming more and more complex
  - Interoperability at stake!
- We need a more powerful framework!
HTTP - the Next Generation

- The generic application level protocol
  - A simple, extensible framework
  - Protocols become Profiles

- Explicit Layering and modularization
  - Break up the big “lump” style 1.x messages

- Extensibility at the core
  - Lessons from our HTTP/PEP/Mandatory work

- Not quite RPC, not quite messages
  - Who cares?
What a Framework Doesn’t Solve

- Will distributed extensibility lead to chaos?
- How do extensions coexist?
  - Two extensions land on the same agent - can they speak, are they orthogonal or do they conflict?
- Central Registration has its place
  - Extension Policing required
- Granularity of Extensions Important
  - Will probably be lumped together in bigger packets
More Information on the Web

- HTTP-NG Project
- HTTP-NG Activity
- HTTP-NG Working Groups (W3C Members only)
- HTTP/1.x Overview
- W3C Member Site
- W3C