Open Pluggable Edge Services (opes)

56th IETF Meeting

San Francisco, CA, USA

Agenda

- Introduction, minutes taker, blue sheets, agenda bashing [5 min]
- Status of WG documents [M. Hofmann, 10 min]
- Start on OPES protocol work
  - Introduction to protocol work [M. Hofmann, 10 min]
  - Protocol layering and the OPES callout protocol [H. Orman, 15 min]
  - OPES scope clarifications [A. Barbier, 20 min]
  - Major decision points for protocol design [A. Rousskov, 15 min]
  - OPES protocol pre-draft thoughts [A. Rousskov, 30 min]
- General discussion
  - Thoughts on using SOAP/XML [A. Barbier, 10 min]
  - Open discussion

Status of WG Documents

- OPES architecture (draft-ietf-opes-architecture-04).
  - Approved by IESG as informal RFC (2/11/2003).
  - Currently in RFC Editor queue.
- OPES protocol requirements (draft-ietf-opes-protocol-reqs-03).
  - Approved by IESG as informal RFC (2/11/2003).
  - Currently in RFC Editor queue.
- Policy requirements (draft-ietf-opes-authorization-02)
  - Submitted to IESG (2/11/2003); waiting for feedback.
- OPES threats and risks (draft-ietf-opes-threats-02)
  - Submitted to IESG (2/11/2003); waiting for feedback.
- OPES scenarios and use cases (draft-ietf-opes-scenarios-01).
  - Previously submitted to IESG, but on hold until threats/risk draft is submitted.
  - Allison Mankin will provided feedback after IETF 56 meeting.

Start of OPES Protocol Work

56th IETF Meeting
San Francisco, CA, USA

What Needs to be Done?

- The WG needs to address two different aspects related to "protocols":
  - An *OPES callout protocol*, which includes
    - the protocol core,
    - (possibly) a mechanism/framework for protocol extensions.
  - *Relationship to application message protocol* (e.g. definition of extension headers for OPES bypass, tracing, debugging).
- Good progress has been made on fundamental design issues for the callout protocol core (see discussions on the mailing list).
- We need to intensify work on
  - Relationship to application message protocol,
  - Tracing/debugging considerations,
  - Security considerations.

Approach to Protocol Work

- Start with a discussion of major decision points that affect the core protocol design, for example:
  - What type of interactions need to be supported (e.g. strict request/response, or request with multiple replies, or...),
  - What type of data is exchanged in requests and replies.
- Based on this discussion, design the fundamental protocol mechanisms and messages in a generic format.
- Follow-up with a discussion of specific protocol bindings/mappings.
  - Should we use binary vs. text vs. XML; should we base the protocol on BEEP or SOAP or HTTP?
- In general: The callout protocol SHOULD be application-agnostic, with support for HTTP being mandatory:
  - We MUST support HTTP as an application protocol,
  - We SHOULD support more than just HTTP,
  - In case of a design conflict or unacceptable complexity, HTTP-arguments have higher priority than arguments for other application protocols.

Rules of the Game

- We are working within the architecture outlined in the OPES architecture draft - this is *not* the time to discuss complex architectural extensions!
  - Deviations from the outlined architecture must be conscious and necessary, and clearly fall within our charter.
- Be specific – If new "requirements" or design issues come up, clearly explain why it cannot be solved within the current environment.
  - Point out a specific desirable behavior that the current OPES architecture, OCP requirements, and pre-draft protocol do *not* allow you to do, and why it still falls under our current charter.
- Let's stick to the terminology introduced in the existing WG documents!
- Always have the IAB considerations in mind.