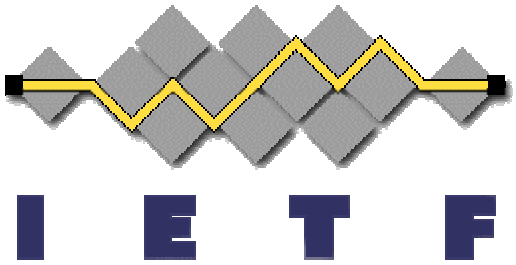


Summary of the ICAP Discussion

55th IETF Meeting in Atlanta, Georgia
(November 17-21, 2002)

Frank Berzau
Technical Evangelist
webwasher.com AG

Introduction

- What's the purpose of this WG discussion
  - "The working group will consider the ICAP protocol drafts as an OPES precursor and will support development of an analysis that explains the limitations of ICAP, to accompany informational publication of that protocol"
  - Consider ICAP as possible protocol candidate for OPES
  - Discuss limitation of current ICAP specification
- submitted ICAP specification is an individual submission and intended to document "current practice"

Agenda

- History of ICAP
- Limitations of current implementation

More Info

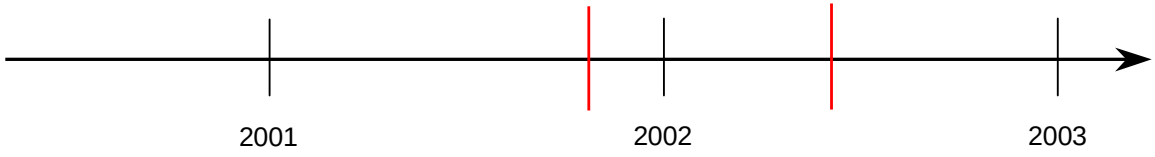
- ICAP Forum
  http://www.i-cap.org/
- webwasher.com AG
  http://www.webwasher.com/

History of ICAP

- ICAP/0.95
- ICAP/1.0
- ICAP Implementations
- RFC Status

Version History

- ICAP/1.0 draft more-or-less stable since June 2001
  - only very few corrections in wordings, explanations and examples
- First shipping ICAP/1.0 implementations since Summer 2001
- draft-elson-icap-00.txt published October 11, 2001
- draft-elson-icap-01.txt with little styling issues published June 7, 2002 (expires in December)



Motivation (general)

- Not every proxy-cache vendor can/want to talk to all service providers and vice versa
- Proprietary interfaces require enormous development and maintenance resources
- Existing ideas were somehow complicated (based on TCP level) while people were looking for a lightweight application protocol
- For customers:
  Chance to exchange some items without reorganizing the complete network infrastructure

Initiators of ICAP

- Akamai
- Lucent
- Network Appliance
- Novell
- WebWasher

Motivation (from 0.9 to 0.95)

- ICAP/0.9 was just like HTTP and original headers have just been added as values of other headers like "ICAP-Modified-Header"
- Complicated and problems with quotation resulted in version 0.95 which was not as open as ICAP should be

Motivation (from 0.95 to 1.0)

- ICAP/0.95 had already an own ICAP header and encapsulated headers following but still allowed chunked transfer encoding only as an option.
- The specification never made it into an IETF draft. Dangerous that this could have resulted in proprietary interface again.
- ICAP/1.0 fixed some problems that have been found with the earlier versions, got its own ICAP identifier (instead of HTTP in request/response) and its IANA reserved port number. It is an open draft from the very beginning.
- At webwasher.com:
  ICAP/0.95 is no longer in use (last installation has been upgraded to ICAP/1.0 this year)

ICAP Implementations (Client)

- ArrayNetworks
- Blue Coat Systems
- iMimic
- Lucent/Bell-Labs
- Microdasys
- NetApp
- Squid (HP Labs/webwasher.com)
- webwasher.com

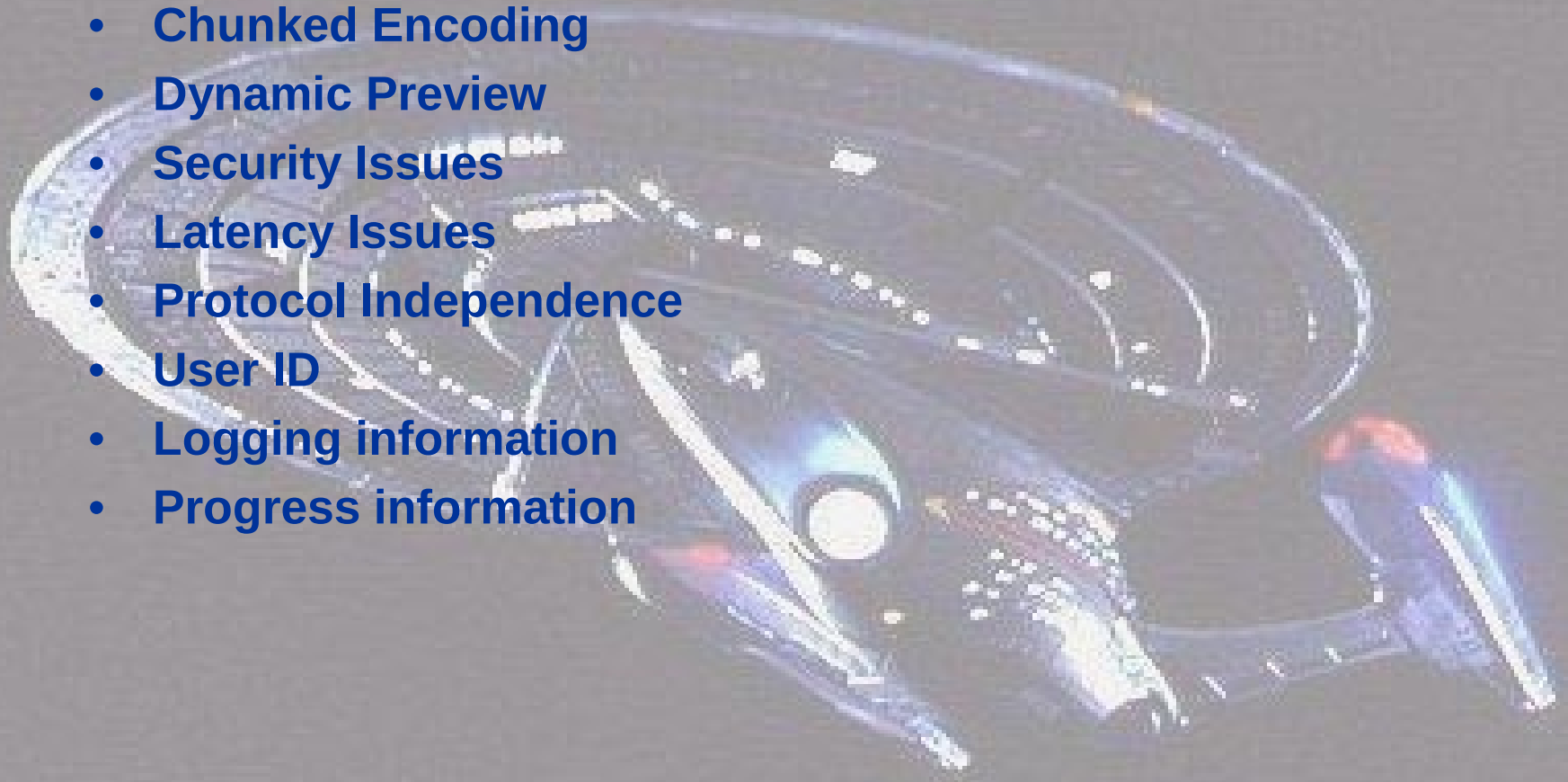
ICAP Implementations (Server)

- Eurecom and Thales (with EU project)
- Finjan
- HP Labs (Python example)
- Lucent/Bell-Labs
- Symantec
- Trend Micro
- WebWasher

Summary

- Draft submitted October 2001
- draft-stecher-opes-icap-eval-00.txt submitted June 2002
- Plan to publish ICAP specs as individual RFC now

Limitations of Current Implementation



- Chunked Encoding
- Dynamic Preview
- Security Issues
- Latency Issues
- Protocol Independence
- User ID
- Logging information
- Progress information

Chunked Encoding

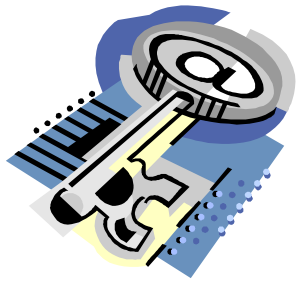
- Byte size and offset values in the encapsulated and preview headers of an ICAP message generate ambiguities between their values and the position of empty lines in the data
- Preperation of ICAP message is more complicated than necessary because we need to wait until encapsulated headers are complete so we can calculate the size, then paste this into the header
- Recommendation
  - Remove the Encapsulated and Preview ICAP header
  - Transfer the complete message in chunked encoding
  - Use chunk extension to mark encapsulated headers and the last preview chunk
  - Each encapsulated header has to be transferred in its own and in exactly one chunk

Dynamic Preview

- Recommendation to support multiple (dynamic) previews
- Send ‚100 Continue' response with next preview size
- Send zero chunk to signal stop sending more previews
  - zero chunk from server - stop it
  - zero chunk from client - ack

Security Issues

- Authenticate ICAP communication
  - Could use shared secret
  - Much like in WCCP v.2
- IP address based ACL on ICAP Server



Latency Issues

- Late clearance trickling must be done by the ICAP Client
- Advantage: Data does not need to be sent back from Server to Client
- Recommendation: New ICAP return code with progress indication
- This would work only for objects that will either be blocked or allowed without modification (final 204) like with antivirus scanning, not with services that actually modify the object (like language translation)
- Progress requests (keep-alive, after data is sent completely, but server is still busy, e.g. unpacking very large archive for virus-scanning)

Protocol Independence

- Today HTTP, FTP and SMTP available in commercial implementations
- Planned NNTP, IM, P2P, Streaming
- Problem SMTP: single sender, multiple recipients = differently modified objects
  - Workaround:
    - Using profile method to get list of profiles per recipient
    - Then send multiple requests according to number of profiles

User Information

- http://www.i-cap.org/spec/draft-beck-opes-icap-subid-00.txt
- Information in ICAP X-Headers
  - X-Authenticated-User
  - X-Authenticated-Groups
  - X-Client-IP
- Need to document/agree on syntax and make this part of the protocol specs

Logging

- Additional logging fields (e.g. Category, Profile)
- Transmitted in additional ICAP response headers
- Needed to consolidate logging at ICAP Client

Summary

- ICAP is available in several commercial applications
- The specs are clear and detailed enough
- ICAP has proven to be reliable and scalable, even in very large deployments