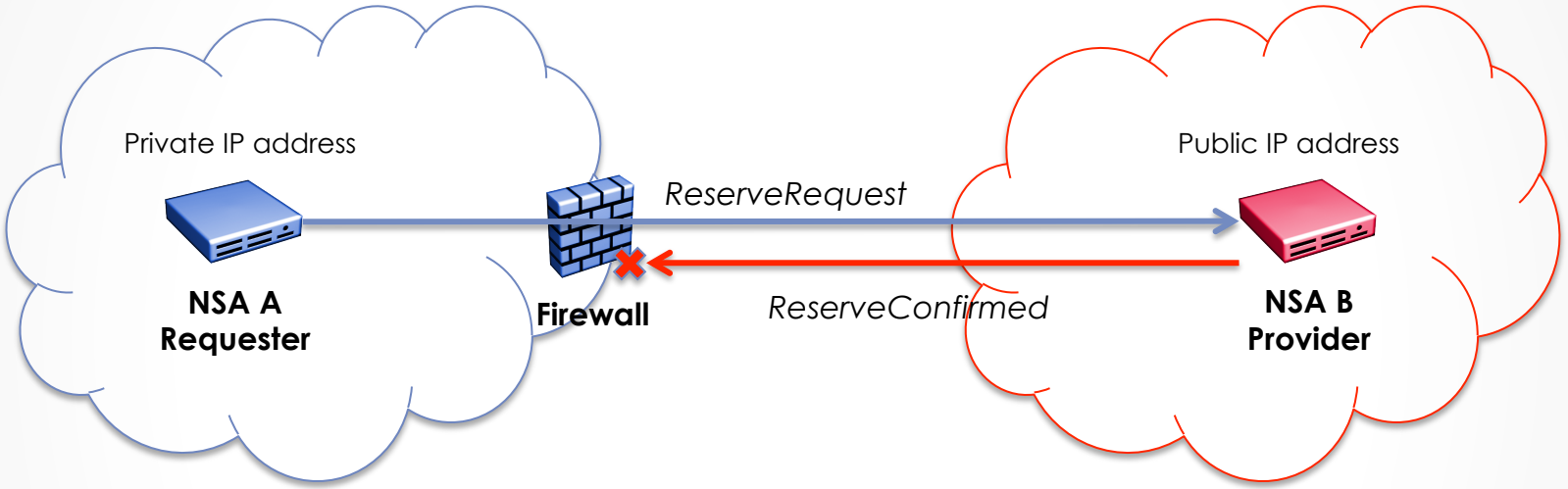
NSI Implementation Task Force

NSI protocol and the dreaded firewall

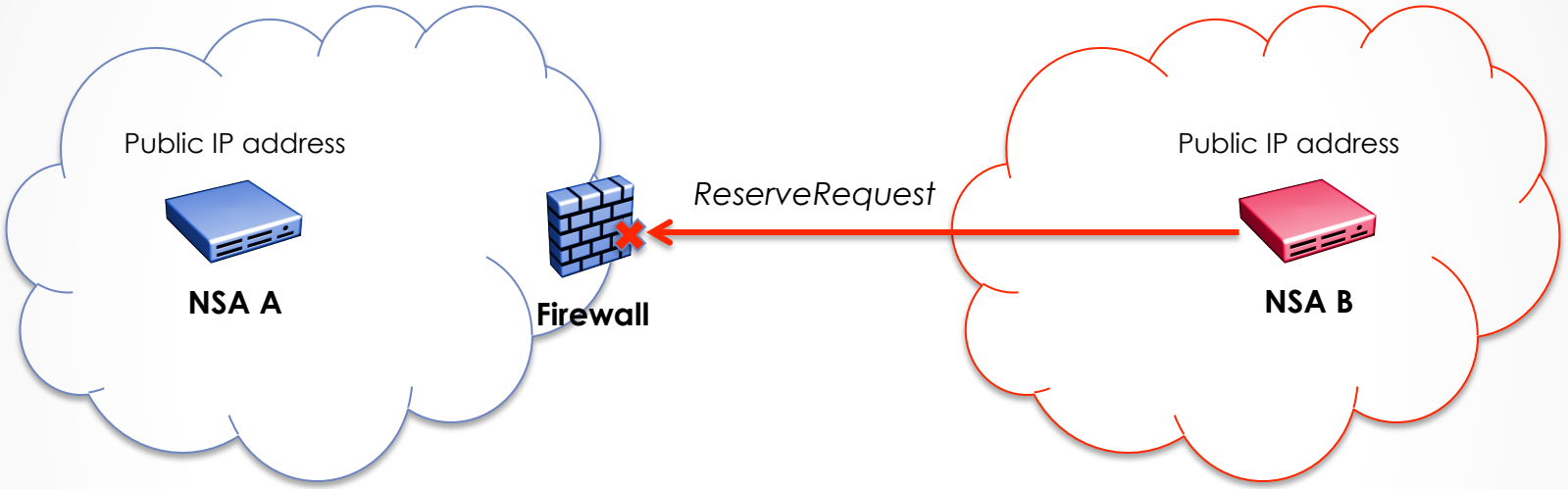
John MacAuley

The Firewall Problem



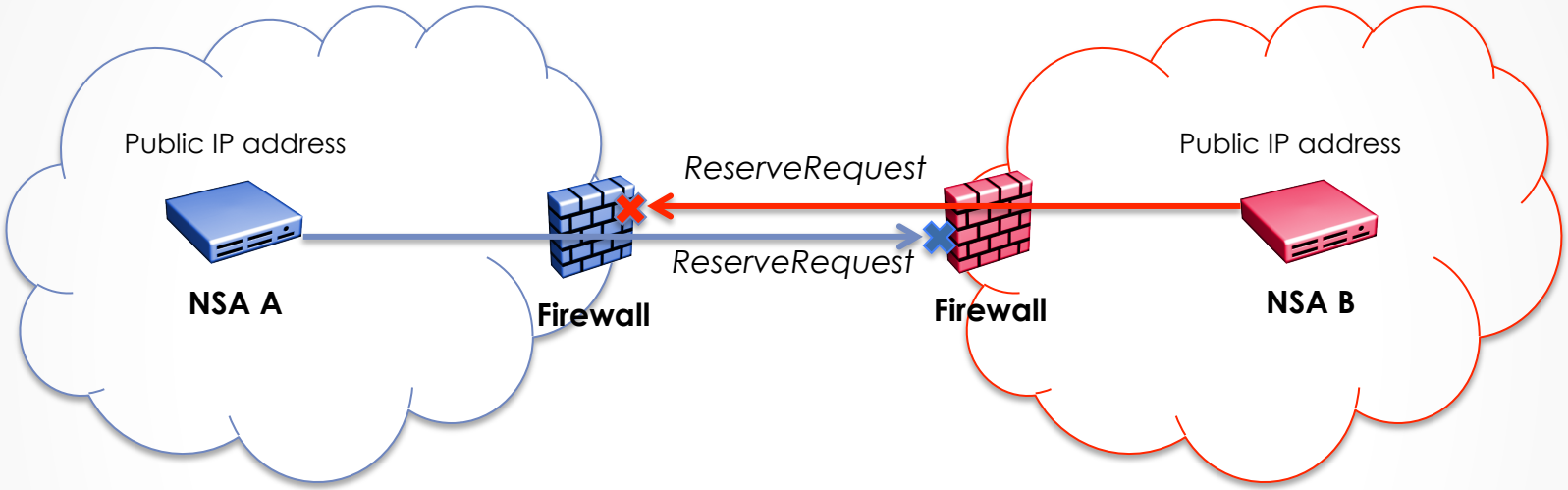
- "NSA A" behind firewall issues HTTP ReserveRequest to "NSA B" on the public network.
- "NSA A" populates private IP address into replyTo field for response.
- Firewall NATs HTTP request and passes on to "NSA B" but does not NAT the private IP address since this is embedded in the SOAP message.
- "NSA B" cannot reach the private IP address to deliver the ReserveConfirmed message.

It Gets Worse



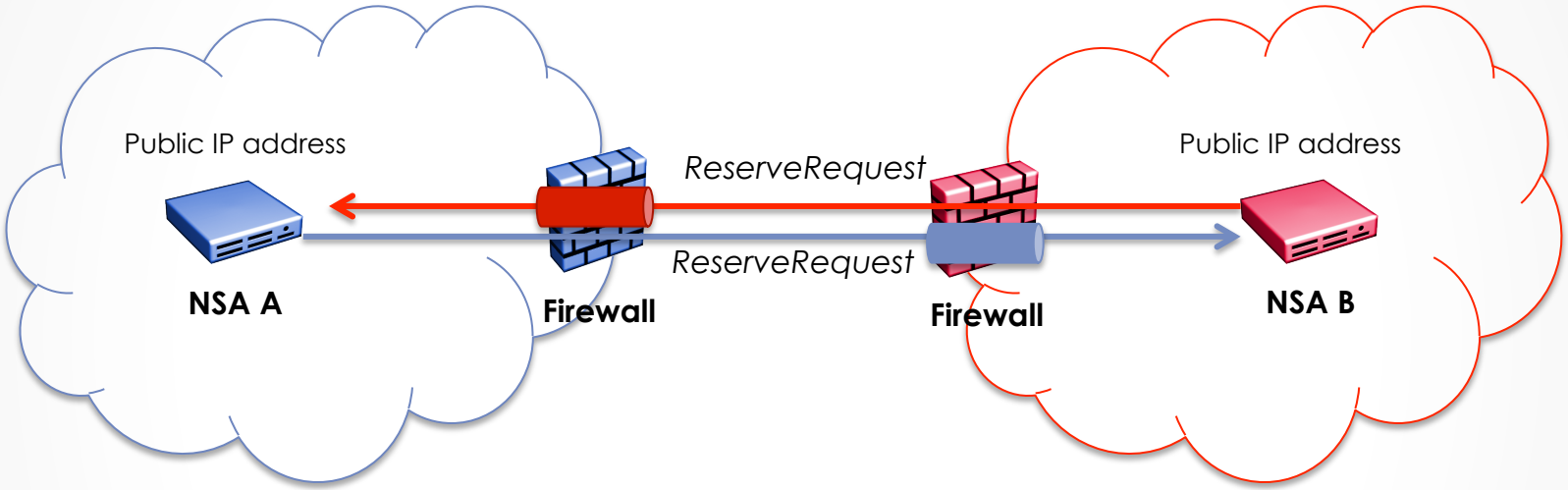
- The problem exists even when public IP addresses are assigned to an NSA behind a firewall, if that firewall is blocking HTTP requests.

Pathological Case



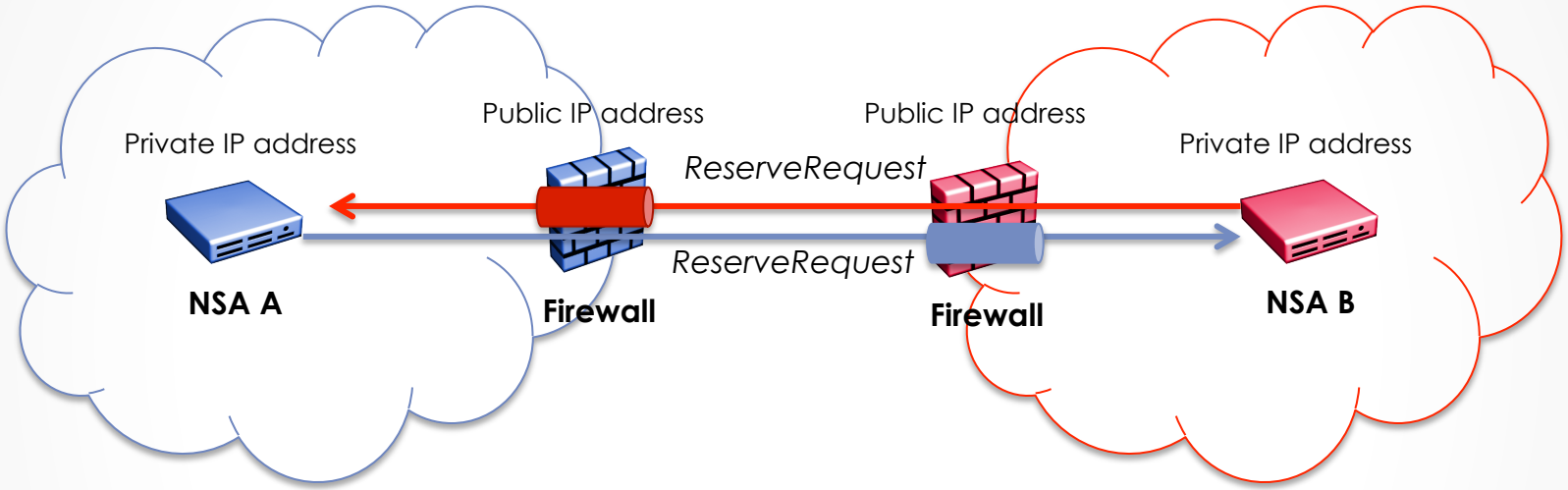
- "NSA A" cannot reach "NSA B" behind the firewall to issue request.
- "NSA B" cannot reach "NSA A" behind the firewall to issue request.

Proper Configuration of Firewall



- For NSA with public IP addresses behind the firewall:
  - Access control lists can be set for peer NSA in combination with port filtering to allow an NSA behind a firewall to have traffic to it's HTTP server port passed through.

Proper Configuration of Firewall



- For NSA with private IP addresses behind the firewall:
  - Firewall will need to act as public entity for NSA.
  - Access control lists can be set for peer NSA in combination with NAT and port forwarding to allow the requesting NSA to be mapped through to the provider NSA's HTTP server port within the DMZ
  - A requesting NSA behind a firewall will need to provide the public facing IP address and port of the firewall within the replyTo field of the SOAP request.

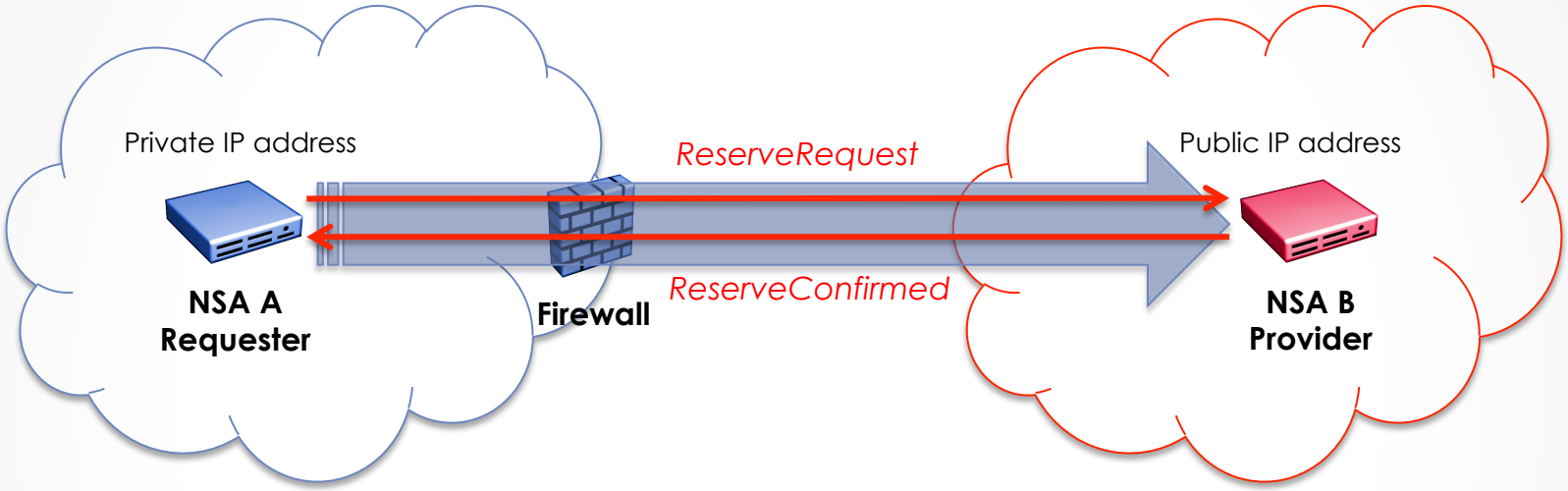
Summary

- Provider NSA must be publically accessible
  - Must have publically accessible interface to receive request messages from Requester NSA.
- Requester NSA must also be publically accessible
  - Must have publically accessible interface to receive response messages (confirm, failed, or event) from Provider NSA.

Discussion

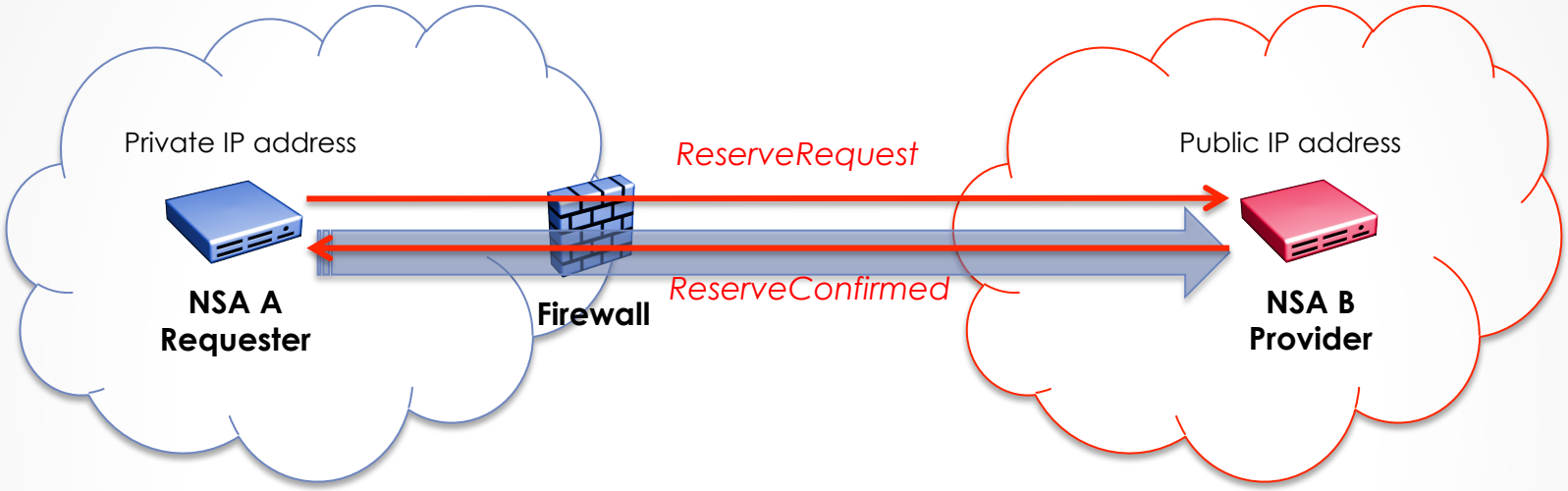
- Can we assume the large majority of deployed NSA will implement both requester and provider roles?
  - Although a good concept for roles in a messaging exchange, does having physically separate concepts of Requester Agent and Provider Agent lead to additional confusion?
- What is a Requester-only NSA?
  - Is this the end user client?
  - Is this delegate for end user requests (if so then it is actually a provider agent from an end user perspective)?
  - Do we administer Requester NSA the same as provider NSA?
- Do we see any value in treating the accessibility requirements of the Requester role differently from a protocol perspective?
  - Should all NSA not be created equal, and therefore, meet the same deployment requirements?

The Dedicated Socket



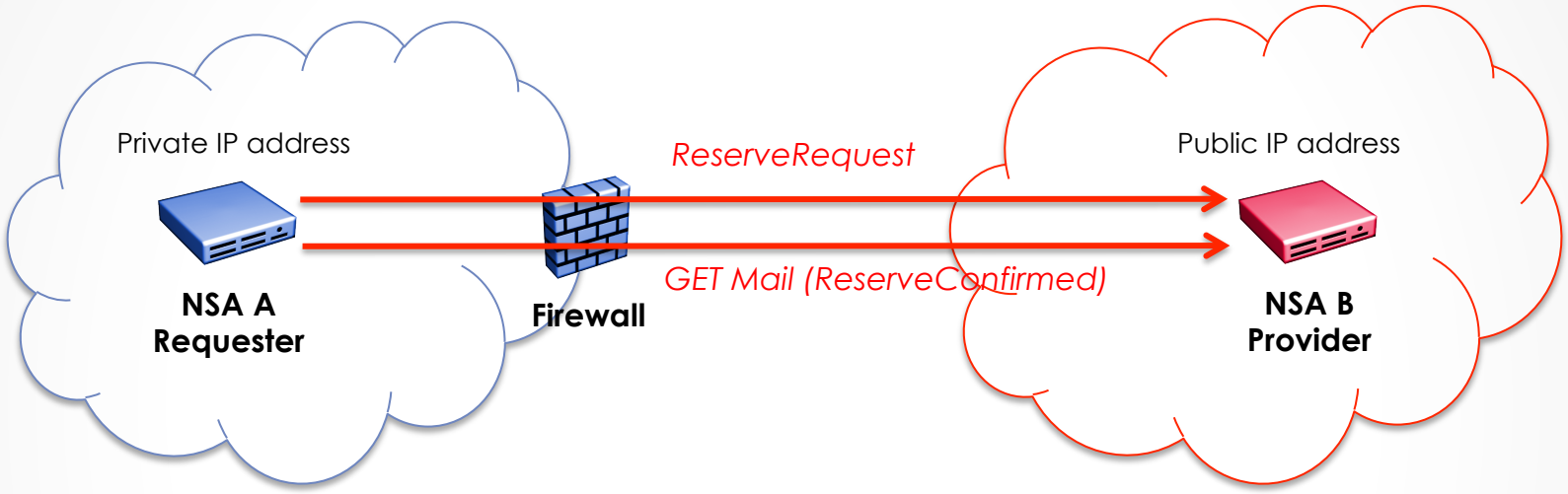
- Introduces a connection oriented transport protocol instead of the current "connectionless" model.
- Provider "NSA A" exposes publically accessible TCP/IP port.
- Requester-only "NSA B" opens a dedicate TCP connection to Provider "NSA A".
- All message exchanges between "NSA A" and "NSA B" occur through this dedicated socket.
- Firewall safe since "NSA B" is creating an out-bound socket request to a publically accessible port, and there is no longer a need for replyTo field.

The COMET Model



- Provider "NSA A" exposes publically accessible HTTP port.
- Requester-only "NSA B" issues requests using standard HTTP request/ response model.
- "NSA B" requests URL for dedicated forwarding discriminator (FD) matching supplied filtering criteria.
- "NSA B" does an HTTP long duration GET on the FD.
- "NSA A" streams messages matching filter criteria (confirmed, failed, and event messages) over the FD HTTP socket.
- Firewall safe since "NSA B" is creating out-bound socket request to a publically accessible port, and there is no longer a need for replyTo field.

The Mailbox Model



- Provider "NSA A" exposes publically accessible HTTP port.
- "NSA A" maintains a "mailbox" for all response and events associated with "NSA B".
- Requester-only "NSA B" issues requests using standard HTTP request/ response model.
- "NSA B" does an HTTP long poll GET on a well known URL.
- "NSA A" sends confirmed, failed, and event messages in response to this long poll (request can block for a short configurable period of time).
- Firewall safe since "NSA B" is creating out-bound socket request to a publically accessible port, and there is no longer a need for replyTo field.

John's Statements

1. NSA are fixed entities that actively participate in the NSI protocol.
2. Clients applications may be (and most often will be) transient entities that come onto and off of the network as needed by the application.
3. NSA are not transient entities since they maintain state of their associated connection reservations.
4. Client applications require a simple and lightweight connection services interface with different requirements and behaviors than an NSA.
5. The NSI-CS protocol as defined for NSA-to-NSA communications is complex, and will only get more complex as more features are added.
6. Application designers don't like complex (they are a simple people).
7. Client applications will most likely be behind one or more firewalls will little ability to get special firewall configurations out of their IT department (application designers are crazy and cannot be trusted).

Proposal

- Leave NSI-CS transport protocol as is
  - An NSA is an NSA so it must be publically accessible.
- Define a simplified client-specific NSI-CS protocol interface
  - A simplified subset of the existing NSI-CS capabilities.
  - Firewall safe.
- This interface should use current best practices for client API development
  - RESTful definition.
  - JSON/XML data representations.
  - Useable from web-based clients.