XCAP

Jonathan Rosenberg

dynamicsoft

Agenda

- XCAP Main spec changes
- XCAP Main spec open issues
- XCAP Package changes
- XCAP Package Open Issues
- Authorization policies changes
- Authorization policies open issues
- List changes
- List open issues

XCAP Changes

- MIME type for PUT/GET of XML elements is application/xml-fragment-body
  - Did not include proposed “root” attribute to define MIME type of actual element
  - Reference XML fragment specification for definition of a fragment
- MIME type for attribute PUT/GET is application/xml-attribute-value
  - Contains only the value, not the name

XCAP Changes

- Merged replace/create subsections
- Clarified that PUT where parent doesn’t exist is an error, returns 409
- Defined default auth policies for documents in the global tree
    - Read all, Write by privileged users only
- Clarified you cannot select comments, namespace attributes or processor instructions

XCAP Changes

- PUT 200 OK response is empty
- Etags are now constant across the whole document
  - Previous mechanism simply didn’t work
  - Client couldn’t determine etag of parent doc or element after a change, in order to change a different part
- Clarified data dependency behavior
  - Client has to GET the resulting data after a PUT, or use package
  - There is a change in etag
- URI hierarchy is a MUST implement

XCAP Changes

- 409 Body Type defined
  - Indicates error and error-specific data
    - Schema invalid, no parent, invalid fragment or attribute value, uniqueness constraint violated
    - For uniqueness violation, the specific URI is indicated and alternatives can be provided
- Client behavior for looking at 409 body and acting on it is included
  - Handling for other error cases is defined in RFC2616
- Document URI and node selector in URI separated by “/” not “?”
  - GET with query strings may be problematic

XCAP Changes

- Documents can now use schemas not understood by the server
    - Document contains an XML “mustUnderstand” element listing required namespaces
    - Equivalent to SIP Require header field

Open Issue 1: Fragment MIME Type

- Issue: should MIME type be application/xml-frag+xml
  - That is, use the RFC3023 convention for XML MIME types
  - RFC3023 unclear on scope of when to use this
- Proposal
  - +xml implies a document compliant to a specific schema or DTD
  - This is a generic xml content type, similar to application/xml
  - Therefore do not use +xml convention

Issue 2: Etag Scope

- Previously, etag scope was a different one for each XML component
- Now, etag scope is whole document
- Problem
  - Rules out cases where there are multiple editors for one document, each operating over a separate section

Proposal

- HTTP does not mandate how the server computes etags, neither should XCAP
- With XCAP, there isn’t an inherent break point in the hierarchy at the document level
- The “natural” granularity for the etag is inherently application specific
  - Ideal granularity is when the normal case is that a client generally only modifies content within the scope of a single etag
- So, specify that application usages should define appropriate RECOMMENDED scopes, but these are not MUST
  - Consistent with HTTP where it’s a server choice
  - If done poorly in the implementation, worst case is inefficiency – protocol still works

Issue 2: Schema Extensibility

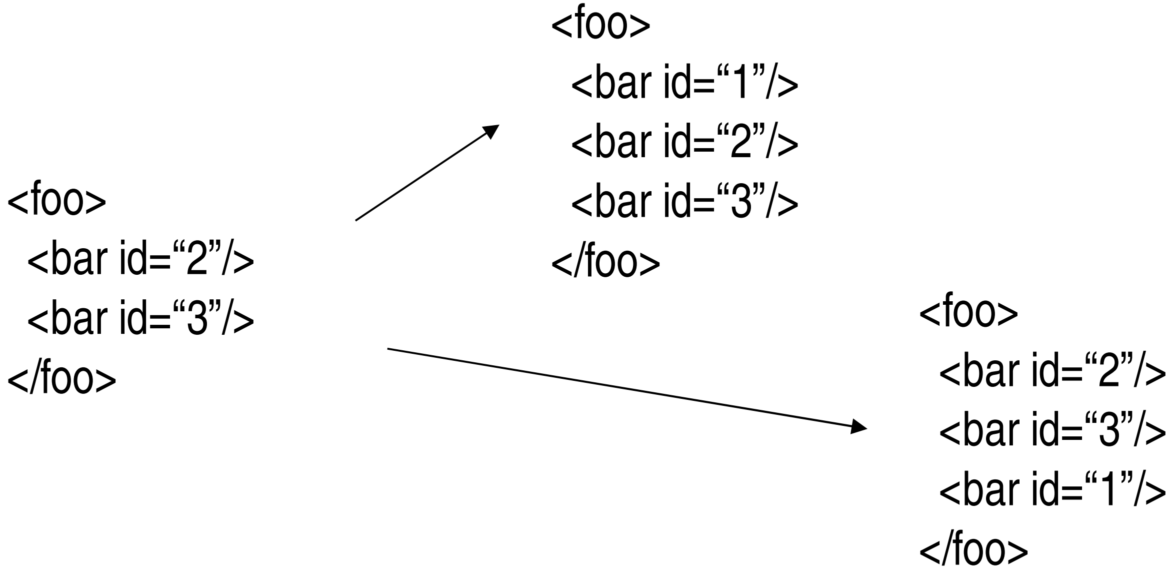
- Current approach is like Require
  - Each document uploaded by the client lists any required namespaces
- Jari proposed an OPTIONS-like approach
  - Define an app-usage that contains “supported-namespaces” in the global tree
  - If the client wants to upload a document which requires the server to understand, it checks this file first
  - If not supported, client does something different

Tradeoffs

- Jari’s approach moves the compatibility check to the client, current one has it in the server
  - Both cases rely on the client and server to properly function
- Jari’s approach has the check out-of-band, current one is in-band
  - In-band includes protocol “ugliness” within documents
  - Server upgrade cases vary
- Server upgrade, in-band implementation
  - Client finds out when server is upgraded only if it tries with extension
  - Trying results in error/retry cycle if there has been no upgrade
  - Not trying delays discovery
- Server upgrade, out-of-band implementation
  - Client can subscribe to list of supported namespaces
  - Will learn when it changes
  - No retrying needed
- Jari’s approach similar to ACAP
- Proposal
  - Adopt Jari’s approach
  - Include the application usage definition inside xcap spec

Issue 3: Insertion Point

- Currently, XCAP does not mandate where an element is inserted when multiple insertion points are possible
  - PUT http://example.com/doc/foo/bar[@id=“1”]



```
<foo>
  <bar id=“2”/>
  <bar id=“3”/>
</foo>
```

```
<foo>
  <bar id=“1”/>
  <bar id=“2”/>
  <bar id=“3”/>
</foo>
```

```
<foo>
  <bar id=“2”/>
  <bar id=“3”/>
  <bar id=“1”/>
</foo>
```

So what?

- Complicates change notifications in xcap-package
- Complicates subsequent ops after PUT
  - Client can’t know position of new element
  - New element position might renumber positions of existing elements
  - A positional selection after such a PUT will be useless
- Proposal: Mandate insertion at the end
  - Doesn’t re-index previous elements!! Very nice.
  - Keeps it simple

Issue 4: Other selectors

- Is the current set of selectors enough
  - Element by name
  - Element by value of its attribute
  - Element by position
- Primary problem is multiple siblings with the same name

```
<list>
  <entry>a</entry>
  <entry>b</entry>
</list>
```

Multi-Name Case

- Positional selection now much more powerful with resolution of previous issue
- GET and DELETE can easily target any element
- PUT for modification can easily target any element
- PUT to create at end is easy
  - N current elements
  - PUT http://example.com/foo/bar[N+1] always inserts
- Only problem case: Insertion into a specific spot

Problem or not?

- Can we mandate that all XCAP schemas do not assign semantics to sibling ordering?
  - Not if we ever need to include an existing schema that has this problem
  - Example problem case: CPL
  - Likely for any other XML domain specific languages
  - Unlikely for XML database types of schema
    - Row ordering irrelevant in relational DB
    - E.g., a non-issue for xcap-cpcp
- No easy way to fix this in XCAP model
- For CPL, can PUT/GET larger pieces or whole CPL
- Proposal: Don’t try to solve this
  - Do not add any additional selectors
  - Add text emphasizing utility of positional selectors

Issue 5: Multiple Insertions

- XCAP allows for insertion or modification of a SINGLE element or attribute at a time
- Implications
  - Adding multiple buddies requires multiple operations
  - Adding multiple users to a dialout conference list requires multiple operations
- For a protocol engineered to manipulate lists, this is a serious limitation

Proposed Fix

- Allow for insertion, modification, fetching or deletion of multiple elements of the same name that are all siblings of the same parent
  - Great for list manipulations
  - Will not be useful for other operations
- How? Easy
  - HTTP URI can use natural Xpath techniques to select several elements
  - For GET and DELETE, result is obvious
  - For PUT
    - If the URI matches no elements in the doc, its insertion at end
    - After insertion, URI MUST reference elements that were present in the body
    - If URI matched some elements in the doc, those are removed and replaced in-place
      - Number of elements in body must match number of elements selected by expression
      - Expression must point to new elements when evaluated

Selecting Multiple Elements

- Introduce Xpath union (|) operator within Predicates

```
<foo>
  <bar id=“1”>A</bar>
  <bar id=“2”>B</bar>
  <bar id=“3”>C</bar>
</foo>
```

Delete Multiples

- DELETE http://example.com/doc/foo/bar[1|2]

```
<foo>
  <bar id=“3”>C</bar>
</foo>
```

Insert multiples

- PUT http://example.com/doc/foo/bar[4|5]
   <bar id=“4”>D</bar>
  <bar id=“5”>E</bar>

```
<foo>
  <bar id=“1”>A</bar>
  <bar id=“2”>B</bar>
  <bar id=“3”>C</bar>
  <bar id=“4”>D</bar>
  <bar id=“5”>E</bar>
</foo>
```

Modify Multiples

- PUT http://example.com/doc/foo/bar[1|2] <bar id=“1”>AA</bar>
  <bar id=“2”>BB</bar>

```
<foo>
  <bar id=“1”>AA</bar>
  <bar id=“2”>BB</bar>
  <bar id=“3”>C</bar>
</foo>
```

Proposal

- Add this capability to XCAP
- NOTE: Not sure on syntax; seems to work according to spec but doesn’t work in XML Spy

Issue 6: Directories

- Important for a client to learn about the documents it owns
  - Bootstrapping for endpoints
  - Determine set of available auth policies for a presence server
- Proposal on list
  - Define an application usage that provides list of documents and their etags
  - And/or use package to subscribe to all documents owned by a user
- Do we need both?

XCAP Package Changes

- Notifications contain etags
- Subscriptions to documents in the global three through a well-known username “global-xcap-user”
- Pending: Allow for subscriptions to all docs for a user

Issue 1: Scope

- Scope 1
  - Only find out the doc changed
  - Effectively a subscription to the etags
- Scope 2
  - Subscribe to change log
  - Find out what the change is, but initial NOTIFY only gives initial etag, not actual document
- Scope 3
  - Subscribe to document
  - Initial notify contains full document
  - Subsequent notifies contain change

Pros/Cons

- Scope 1 is the simplest, ideal for where a single user edits their own doc as the normal case
- Scope 3 is general purpose, more complex, overalps a bit with XCAP itself
  - HTTP GET or initial SUBSCRIBE return full doc
- Not clear scope 2 buys much over 3
- Proposal:
  - Scope 3
  - Add package parameter to ask just for etags

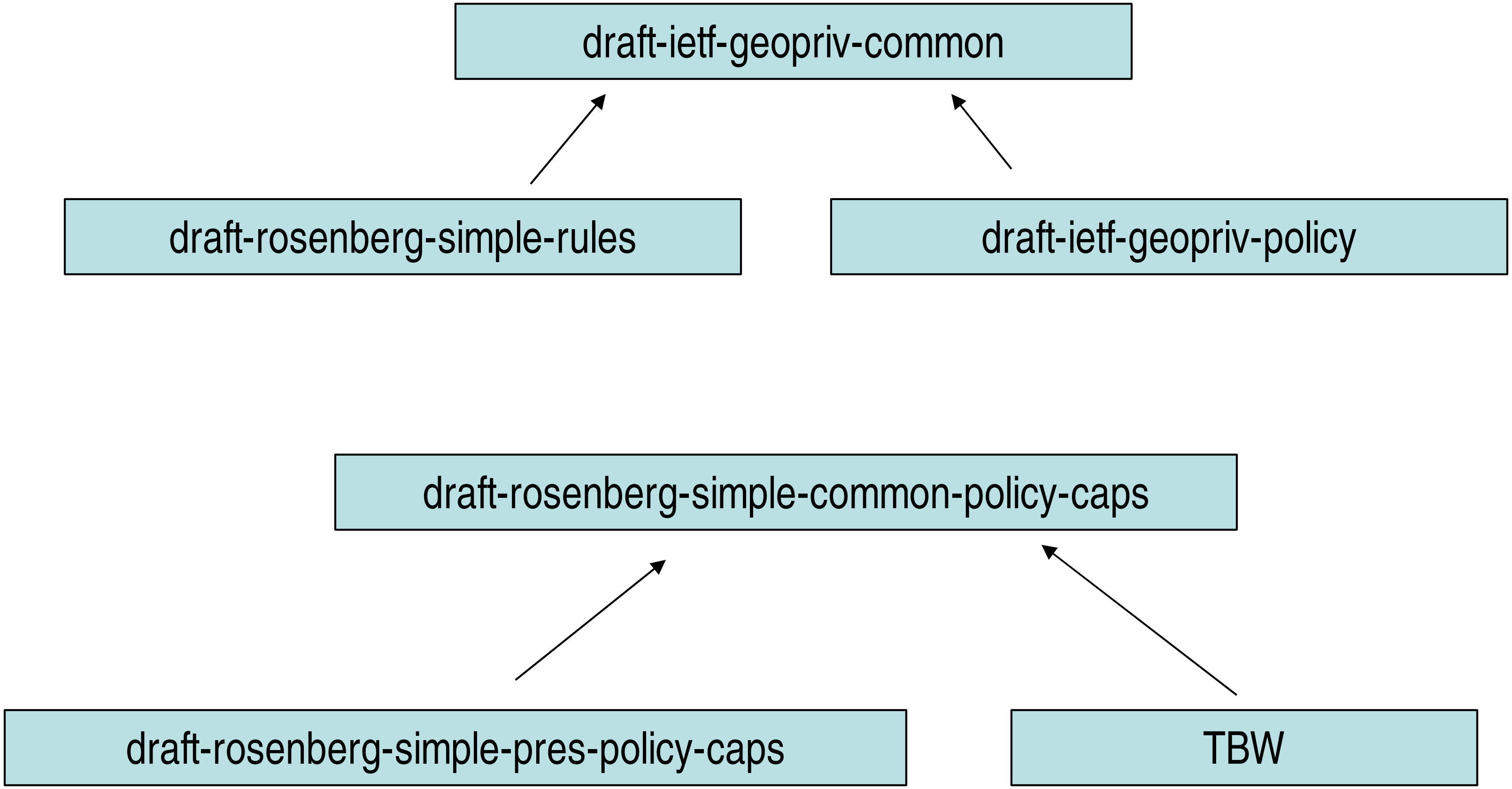
Issue 2: Deterministic Changes

- Change notification format doesn’t specify where an insert occurs
- With current XCAP, server and client may compute different documents
- This is resolved with previous XCAP proposals

Issue 3: Config Framework

- Should we align xcap package with SIP configuration framework?

Authorization Changes



draft-ietf-geopriv-common

draft-rosenberg-simple-rules

draft-ietf-geopriv-policy

draft-rosenberg-simple-common-policy-caps

draft-rosenberg-simple-pres-policy-caps

TBW

Authorization Changes

- All conditions are part of a <condition> element
- New <validity> condition
- Only domains have <except> clauses
- <sphere> condition
- Removed <can-encrypt> condition
- Explicit subscription confirmation action
- Explicit polite blocking action
- Explicit rejection of subscriptions
- Removed <encrypt> action
- <anonymous> a global condition
- Three xforms – show tuple, show-namespace, show-element
  - Each applied independently
  - Each takes a pass at removing data
  - Unfortunately they overlap in coverage
- Less xcap centric

Issue 1: Semantic v. Syntactic

- Current policies are syntactic oriented
  - Can specify policies for PIDF elements not yet defined
- However
  - Overlap in which XML components are selected by each policy introduces complexity
  - Certain policies are not easily expressed syntactically
  - Mapping from syntactic policies to UI may be complex
  - Easy for rules to create invalid PIDF documents
  - Attribute restrictions would introduce sizeable XPath complexity [?]

Proposal: Semantic

- Include basic PIDF policies
  - Control access to note
  - Control number and types of tuples
- Include RPID policies
  - Primarily hide or show each attribute
  - Possibly globally or per tuple using class
- Include guidelines for other PIDF extensions to define their own policies
- Specific details on list shortly

Presence List Changes

- No authorization specified here about who can subscribe to the list
- Display name optional
- Entry URI mandatory, id optional
- Added <entry-ref> which points to an entry elsewhere in the list
  - Allows one buddy to appear on multiple lists without repeating information

Issue 1: Other List Source

- In many systems, some other list (possibly non-XCAP) will serve as the real “address book”
  - Enterprise directories
  - Wireless phone book
- In such a case, most information on users resides there
- Presence list need only contain flat list of URIs for the presence list
  - No structure needed
  - No auxiliary data needed – display name, etc.
- Client needs to know whether it should put structure and aux data into presence list or not
- Proposal: Define a global document that includes such an indication

Advanced IM Requirements

- Outlines requirements for new work to cover
  - IM delivery notifications
  - IM “is typing” indicators
  - IM receipt capabilities
  - Group page mode
  - Invitations to non-real-time sessions
- Most discussion on mechanisms for IM delivery
- Main question – are we still interested in each of these?