Revisiting name-content binding: In-network namespace operations

breakout session
room Melnitz 1410 (scoring stage)

NDNcomm2015, UCLA, Sep 28+29, 2015
Christian Tschudin
When things were simple...

- **CCN Names** identify an information collection (not an information container).
- Name hierarchy indicates membership.
- The same information can have many names (web-like links).
- Like IP, a CCN node imposes no semantics on names — meaning comes from application, institution and global conventions reflected in prefix forwarding rules.

```
App supplied name | Versioning & segmentation | Content or proxy (e.g., SHA256 checksum)
```

```
/parc.com/van/cal/417.vcf/v3/s0/0x3fdc96a4...
```
.. and how developers use names

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nameless object</td>
<td><em>a “name” being just a digest</em></td>
</tr>
<tr>
<td>overriding names</td>
<td><em>redirection, LINK object, NDNS</em></td>
</tr>
<tr>
<td>named functions</td>
<td><em>lambda expressions with many names</em></td>
</tr>
<tr>
<td>names with blanks</td>
<td><em>unifyable names, wildcards</em></td>
</tr>
<tr>
<td>attribute sets</td>
<td><em>instead of hierarchy, data warehousing</em></td>
</tr>
<tr>
<td>names expressing time-dependent content proximity</td>
<td><em>InfoMax</em></td>
</tr>
<tr>
<td>crafting certification chains into the data namespace</td>
<td><em>schematized trust</em></td>
</tr>
<tr>
<td>encrypted names, one-time names, …</td>
<td></td>
</tr>
</tbody>
</table>
Goal of this breakout session

A. Do we have a name crisis?
   - collecting more evidences of “wild” names
   - review the list we have so far

B. Discuss “who has to care”
   - network?
     (name portion for routing, but also clever caching, certs)
   - storage side?
     (name portion being a database query)

C. Discuss network-assists for namespace operations
   - name rewriting (redirection, expression expansion)
   - namespace stitching (filesystem names, service chains)
Where this could lead … (personal view)

- **Fight the overloading of names: disentangle concerns**
- “name neutrality test”: does your app continue to work if name components are reshuffled?
- “name futureproofness test”: does your app work in 10 years (changed providers, expired certs …)
- Think in terms of a “mount” operation (from Van’s talk on set reconciliation: purpose matters, not method)
  - “publishing” is exporting a namespace
  - implies rewriting
- **Lowlevel corollary:**
  - Interests “with name slots”
  - multiple signatures (per carrier, per app, per stitching etc)
Actual discussion

• New LINK object (redirection) triggers quite some discussion
  - security concerns
  - **agreement that functionality is desirable**, but methods to be studied and scrutinized

• Classic CCN&NDN naming does not provide ID/Locator separation as names are topologically significant and used for routing
  - problematic for producer mobility, for switching provider

• Audience has interest in name-less objects and **manifests**:
  - Nacho explains: Manifests are explicit data structures to enumerate names belonging to a collection
  - manifest-signing instead of signing each piece, pre-caching
  - replaces implicit model where name “stands for” a collection