certspec

Sean Leonard, Penango, Inc.
IETF 84 APPSWG
Monday, July 30, 2012
What is certspec?

urn:cert:issuersn:CN=Atlantis;2A

• Uniform syntax for
• identifying
• a *specific* certificate
• in a textual format
URN Primer

• Resource identifiers that are **persistent, location-independent, text-based** (**transcribable** by keyboard & **recognizable** by humans), **mappable** to other URIs

• RFC 2141; urnbis

• Examples:
  – urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
  – urn:oid:1.3.6.1.4.1
  – urn:ietf:rfc:2141
  – urn:isbn:0-395-36341-1
What’s the ISBN of this book?
Motivation

• Apps
  – in preferences for runtime retrieval
  – for exchange

• Protocols
Use Cases

<?xml version="1.0"?>
<props>
<host>service.example.com</host>
<port>443</port>
<tls enabled="true" minVersion="1.1">
  <sni enabled="true"/>
  <servercert>urn:cert:SHA-1:b1f090a8e2d70353107454f9618347b18b321bf1</servercert>
</tls>
</props>

JSON ("trusted certs")
[“urn:cert:SHA-256:0de4564b5c09c7fbd2a1fade71d5d3ae5613e2e33de49c8f15fec2cafa592f58”,
 “urn:cert:SHA-512:f2d956ab9510adffdd38c26e84f3d2116ec8174190c587ee26147d57bba2dccb2e0e0944ea60086a045d490df6f8648dae673fe66877e05d632efdd3a8c4b1bdb”,
 “urn:cert:base64:MIHUAlgINCAQICASowCQYHKoZIzj0ETAMMMQowCAYDVQQDEwFRMB4XDTEyMDCzMDExMjc0MVVoXDTE0MDczMDEyMjc0MVowDDEKMAgGA1UEAxMBUTBOMBAGByqGSM49AgEBSuBBAAhAzoADObkALyjNzblvjALOb1mHIlqQnpJGBGaKqmLgK1silgLAIbMcMaVdVvwR6leSNVF/PnV02qTRi j6YKMAkGByqGSM49BAEDPwAwPAICG6jgr8tVG6un50rqHuN48ZxzRYQjfJnuSNzpTwIcETJpAVPSdk3Yz2 evgSfZktTpfl8vkJvLiEcHzA=""]
Features of Certs

• Standardized objects (X.509, PKIX)
• Have canonical encoding (DER)
• Variable size (in-band or OOB may be better depending on application)
• Have a hierarchical namespace (issuer + serial number) or can be identified by exactly one hash*
• Used in security protocols; accurate identification is critical
Mechanisms

```
urn:cert:SHA-256:0de4564b...fa592f58
```

- by-reference
  - by-hash (SHA-1, SHA-2) (not “parameterized”)
  - by-data (issuersn)
- by-value
  - data (base64, hex)
Comparisons & Next Steps
Compare certspec and status quo

- Meets URN criteria
- Existing preferences not portable, exchangeable, or algorithm-agile
- Different protocols reinvent the wheel
- Want by-value and by-ref agility
  - Longest hash (128)
  - Shortest (practical) value (241/329)
    - eliminates DoS vector, lookup time
## Compare certspec and ni

<table>
<thead>
<tr>
<th>certspec</th>
<th>ni</th>
</tr>
</thead>
<tbody>
<tr>
<td>URN</td>
<td>URI</td>
</tr>
<tr>
<td>Canonical encoding</td>
<td>No canonicalization</td>
</tr>
<tr>
<td>Resolves to any URI/protocol</td>
<td>Implies “ni-capable protocol” with specific (but unspecified) behaviors</td>
</tr>
<tr>
<td>Accurate, unique identifier</td>
<td>Not unique</td>
</tr>
<tr>
<td>No truncation allowed (“security”)</td>
<td>Truncation encouraged (“flexibility/brevity”)</td>
</tr>
<tr>
<td>One identifier per URN, not query lang</td>
<td>Multiple identifiers</td>
</tr>
<tr>
<td><strong>Different algorithm considerations</strong></td>
<td></td>
</tr>
<tr>
<td>Limited to certs</td>
<td>Digital things</td>
</tr>
<tr>
<td>Trivial transcription from crypto tools</td>
<td>Full support requires new implementations</td>
</tr>
</tbody>
</table>
Next Steps

• Harmonize with urnbis
• Improve Motivation section
• Discuss extensibility aspects
• Allocate NID

Questions?