OR.NET Strategy Meeting: Transport Layer
2021-11-02
David Gregorczyk
Motivation

- MDPWS is based on outdated standards
  - HTTP/1.1 1999
  - DPWS 2009 – technical committee was closed in 2016
    - Unresolved issues
    - Referenced standards are outdated
    - Fixes are up to the IEEE standards committee; complexity of the protocol does not come with any benefits anymore
- Verification of MDPWS is cumbersome and leads to tons of contradictions as references are not sufficiently harmonized/profiled
- MDPWS is based on XML which is not a proper communication technology for SOAs/ROAs these days
  - In terms of efficiency and commonness
- Many pain points during implementation
Case study

• In 2020, we ran a case study to find a technology better suited for resource-constrained devices
• Protocol buffers / gRPC delivered promising results
  • Cloud Native Computing Foundation project (Linux foundation), big community backbone
  • Global players are using it (Square, Netflix, Cisco, etc)
  • Efficient implementation in different programming languages
  • Upgrades come for free (bug fixes, HTTP 3.0 migration)

• Working title: protoSDC
  • Leverages protocol buffers over gRPC with HTTP 2.0
  • Compatible with BICEPS
    • Eventing, Discovery, Security, Participant Model, Message Model
How does it work?

- A toolset transforms the BICEPS XML Schema (and potentially any XML Schema) into a protoBuf schema
- Further tooling allows to transform between protoBuf and auto-generated custom model classes:
- Services in gRPC are modelled manually
## Performance tests

<table>
<thead>
<tr>
<th>MDPWS</th>
<th>protoSDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>~150 referenced standards</td>
<td>~15 referenced standards</td>
</tr>
<tr>
<td>SDCri throughput of reports/states per 5s: ~1,300/248,000</td>
<td>SDCri/proto throughput of reports/states per 5s: ~86,000/1,650,000</td>
</tr>
<tr>
<td>Based on DPWS; used by Apple, Cisco, Netflix, Dropbox and others</td>
<td>Based on gRPC; used by Google, Cisco, Netflix, Dropbox and others</td>
</tr>
<tr>
<td>Discovery Hello Message: ~1000 Bytes</td>
<td>Discovery Hello Message: ~130 Bytes</td>
</tr>
<tr>
<td>Discovery with limited interoperable support of Discovery Proxy</td>
<td>Discovery proxy support has been included explicitly</td>
</tr>
<tr>
<td>QName usage ambiguous</td>
<td>QName usage unambiguous; interoperability is proved by reference implementations in multiple languages comprising C/C++, C#, Python, Java, … openly available</td>
</tr>
<tr>
<td>Syntactical interoperability hard to achieve; limited amount of implementations available</td>
<td>Syntactic interoperability is proved by reference implementations in multiple languages comprising C/C++, C#, Python, Java, … openly available</td>
</tr>
<tr>
<td>Eventing: at least two connections required to transmit data between peers</td>
<td>Eventing: one connection; leverages HTTP streams</td>
</tr>
</tbody>
</table>

![Graph showing performance comparison](image)
How to proceed

• MDPWS is far beyond the point that it can be easily ruled out / replaced

• But: keep working on implementing protoSDC reference implementations for future migration
  • sdc11073 2.0 will have a gRPC integration (done already, not pushed yet)
    • GitHub - Draegerwerk/sdc11073 at version_2
  • SDCri has a feature branch open that integrates a gRPC binding
    • Files · feature/proto-sdc · sdc-suite / SDC Reference Implementation · GitLab
  • New: Kotlin-based implementation from scratch is about to be finished (async programming and easy integration with Android)
    • sdc-suite / protoSDC-kt · GitLab
  • New: Rust-based implementation from scratch is about to be finished (async programming and easy integration with iOS)
    • sdc-suite / protoSDC-rs · GitLab

• 95% feasible, 5% to be done (e.g., best way to handle extensions)

• Near future: Discuss protoSDC as topic of interest in SDPi and provide a protoSDC option eventually

• Far future: Start with 11073-20701 revision that incorporates a gRPC/protobuf binding
Thank you for your attention!
David Gregorczyk
david.gregorczyk@ornet.org