Remote Vehicle Interaction
October 19, 2016  |  Expert Group Update

Magnus Feuer
Head System Architect  - Open Source Projects, Jaguar Land Rover

GENIVI is a registered trademark of the GENIVI Alliance in the USA and other countries. Copyright © GENIVI Alliance 2016.
Progress Report – Overview

- SOTA and SWM handed over to Network Expert Group
- VSS and VSI handed over to Network Expert Group
- RVI C Library Completed
- PKI Completed
- Provisioning completed
- Big Data Expanded with backend processing
Failure Report – Overview

• GENIVI Compliance process for RVI Core
Progress Report – SOTA / SWM

• SWM picked up by several OEMs and T1s
• Successful OSS -> Commercial transition
Progress Report – VSS / VSI

- 847 signals and counting
- Adopted by W3C and OCF
- Separate session Wednesday 11:00-12:30.
Progress Report – Big Data

• Collaboration with Hortonworks
• Get the bigger picture through data aggregation
• Separate session Wednesday 14:00 – 15:00
RVI C Library

Tatiana Jamison
C applications plug in to RVI architecture
## Simple API for RVI behavior

<table>
<thead>
<tr>
<th>Context management:</th>
<th>I/O management:</th>
</tr>
</thead>
<tbody>
<tr>
<td>rvi_init(…)</td>
<td>rvi_process_input(…)</td>
</tr>
<tr>
<td>rvi_cleanup(…)</td>
<td>rvi_invoke_remote_service(…)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service management:</th>
<th>Connection management:</th>
</tr>
</thead>
<tbody>
<tr>
<td>rvi_register_service(…)</td>
<td>rvi_connect(…)</td>
</tr>
<tr>
<td>rvi_unregister_service(…)</td>
<td>rvi_get_connections(…)</td>
</tr>
<tr>
<td>rvi_get_services(…)</td>
<td>rvi_disconnect(…)</td>
</tr>
</tbody>
</table>
RVI keeps dependencies to a minimum

- OpenSSL
  - GENIVI specification
- libJWT
  - C implementation of JavaScript Web Token standard
- Jansson
  - C implementation of JSON standard
- *Future dependency: MessagePack library*
RVI Core – Under the hood

Ulf Wiger
Current RVI Core Priorities

• Security
• Stability/Maintainability
• Usability
• Speed/Scalability
RVI Core Security

• Security Team review ongoing
• Code cleanup
  – Simpler code
  – Remove unsecure transports/ports
  – Explore advanced use cases
Stability/Maintainability

- RVI concepts have stabilized
- Remove debris from previous versions
- Focus hard on readability
- Maintain test suite

- (rewrite in progress – done in near future)
Usability

• Support key semantics
  – RPC with immediate failure notification
  – Message order guarantees
  – Dynamic provisioning
  – Advanced use cases (e.g. RVI-provisioned tunnels)
Dynamic Provisioning

• Core RVI-related actions
• Provisioning Server logic up to each implementor
• Credential exchange uses RVI protocol

• Works with rvi_core 0.5.0
Dynamic Provisioning Sequence

Credential authorizing 'get_creds' below

How to vet Guest not covered (site-specific)
RPC + Service invocation relay

{cmd: rcv, service_name: S, reply_to: Sx, route: [A], ...}
Message Order Guarantee

- "Messages from endpoint A to endpoint B should arrive in the same order as they were sent"
  - If they arrive at all
  - Provided same "chan" attribute
- 'Minimal' ordering
  - Parallelize as much as possible
  - Enforce ordering when needed
Relay-based connection fan-out
RVI-provisioned tunnel (nyi)

- Could be extended to multiple hops
- Relay service does basic packet forwarding
- Require random cookie as first message?

- socket
- RVI tls conn
Thank you!

Visit GENIVI at [http://www.genivi.org](http://www.genivi.org) or [http://projects.genivi.org](http://projects.genivi.org)

Contact us: [help@genivi.org](mailto:help@genivi.org)