Testing of Connected Functions

Concept for CANoe in the Cloud
<table>
<thead>
<tr>
<th>Agenda</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CANoe Simulation Approach</td>
</tr>
<tr>
<td>2.</td>
<td>Concept: CANoe-in-the-Cloud</td>
</tr>
<tr>
<td>3.</td>
<td>Integration and Testing Approach</td>
</tr>
<tr>
<td>4.</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Simulate the “needed” environment here: model ECU2 and simulate rather than integrate the real ECU2
- High coverage of system states possible
- Variants can be simply supported in one model (“150% model”)
- Less complex
  - Only aspects which are really needed in actual context – which is often rather simple
- No special embedded SW needed (for ECU2)
  - Which often has to have a specific patch/version
  - Which causes a lot of effort to maintain and deploy
  - Where right version of SW needed
- No extra harness which is error prone
- Can be debugged

CANoe approach: Simulate the whole “Remaining” System Environment
CANoe Simulation Approach

Connected SUTs - Analogy from Classic World

- Simulate the needed environment here: model the backend function F2 and simulate

Advantages: see previous slide
<table>
<thead>
<tr>
<th></th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>CANoe Simulation Approach</td>
</tr>
<tr>
<td>2.</td>
<td>Concept: CANoe-in-the-Cloud</td>
</tr>
<tr>
<td>3.</td>
<td>Integration and Testing Approach</td>
</tr>
<tr>
<td>4.</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
**System Overview**

- **Concept:** CANoe-in-the-Cloud

**Functional System**

- COM-Module: (Wifi, xG, BT, ...)

**SUT**

- Classic IF
- “Connected” IF

**Classic Access:**

- CAN, LIN, ETH, I/O, ...

**Internet:**

- 4G, 5G, ...

**Protocol IP based**

- CANoe proprietary IP based: HTTPS, WebSockets, ...

**CANoe (on-PC) - Client**

- Instance 0001

**CANoe-in-the-Cloud**

- Tunnel only or with model
  - Instance 0001
  - Instance 0002
Concept: CANoe-in-the-Cloud

Properties

- Connects to the SUT directly via Internet
- Thus no specific IT policies required
- However, SUT must connect to Vector cloud instead of real productive cloud
- First Step: Provide tunnel from CANoe to connected SUT
- Next Step: Parts of the CANoe models can be uploaded to CANoe-in-the-Cloud
- Every CANoe client instance instantiates one CANoe-in-the-Cloud instance
System Architecture

Concept: CANoe-in-the-Cloud
Concept: CANoe-in-the-Cloud

Cloud Part (Server) - Workflow

1. Create new instance
2. Get endpoint based on port
3. Forward request
4. Request certificate creation
5. Store certificates
6. Request container creation
7. Verify instance exists
8. Get port configuration
9. Invoke shell via SSH
10. Pull docker containers
11. Create and run docker containers
12. Store configuration
13. Forward tunnel connection
14. Create TCP/UDP connection
15. Forward TCP connection

Service Gate
CANoe Mgmt Server
Network Manager
Proxy Server
Tunnel Server
Model
EC2
Certificate Server
Docker Registry
Proxy Server
Tunnel Admin
Tunnel
EC2
Cloud Part (Server)
Cloud Part (Server) - Workflow

1. Tunnel Admin
2. Service Gate
3. Proxy Server
4. Model
5. Tunnel Server
6. EC2
7. Docker Registry
8. Certificate Server
9. CANoe Mgmt Server

Steps:
1. Get endpoint based on port
2. Forward request
3. Request certificate creation
4. Store certificates
5. Request container creation
6. Verify instance exists
7. Get port configuration
8. Invoke shell via SSH
9. Pull docker containers
10. Create and run docker containers
11. Store configuration
12. Forward tunnel connection
13. Create TCP/UDP connection
14. Forward TCP connection
Concept: CANoe-in-the-Cloud

Cloud Part (Server) - Workflow

1. Tunnel Admin
2. CANoe Mgmt Server
3. Network Manager
4. Proxy Server
5. Certificate Server
6. Docker Registry
7. EC2
8. Model
9. Tunnel Server
10. Service Gate

- Create and run docker containers
- Pull docker containers
- Create TCP/UDP connection
- Forward tunnel connection
- Request container creation
- Forward request
- Request certificate creation
- Get port configuration
- Invoke shell via SSH
- Pull docker containers
- Get endpoint based on port
- Verify instance exists
- Store certificates
- Store configuration
- Change access network
- Get data from CANoe
- Send data to SUT
- Create new instance
- Request container creation
- Store certificates
- Invoke shell via SSH
- Pull docker containers
- Get port configuration
- Verify instance exists
- Store certificates
- Change access network
- Get data from CANoe
- Send data to SUT
- Create new instance
Concept: CANoe-in-the-Cloud

Desk Part (Client)

**Tunnel IF & Management**

- **Admin Tunnel** (Tx commands)
- **Data Tunnel** (Tx and Rx Data)

**Concurrent Queue**

- WebAccess
- HTTPS
- WSS

Cloud

Main, Create Instance, etc...
<table>
<thead>
<tr>
<th>Agenda</th>
<th>1. CANoe Simulation Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2. Concept: CANoe-in-the-Cloud</td>
</tr>
<tr>
<td></td>
<td>3. Integration and Testing Approach</td>
</tr>
<tr>
<td></td>
<td>4. Conclusion</td>
</tr>
</tbody>
</table>
Integration and Testing Approach

**Excursus: SUT - Differences in Test Focus**

**SUT = One ECU (classic)**
- Focus
  - ECU as black box
  - Signals on classic networks

**SUT = Distributed Function (new)**
- Focus
  - Software Interfaces
  - Service level
  - Function calls
  - Logical information flow
  - No ECU context

1. $f_1$
2. $f_2$
3. $f_3$
4. $f_4$
Integration and Testing Approach

Excursus: SUT - Differences in Test Focus

Same as in classic approach: Different levels of simulated and real parts of SUT.
Conclusion

For ECU/function developers:

▶ The typical CANoe simulation approach still applies
▶ Possibility to model every aspect rather provide real parts
▶ Bring in the real backed into your simulation
▶ No specific requirements to IT infrastructure because of standard Internet access
▶ The testing workflow stays the same
▶ Even when we look more into SW testing rather than ECU testing

For backend developers:

▶ Concentrate on the backend
▶ All „automotive aspects“ are covered by CANoe
Thank you!

For a live demonstrator please visit booth in the exhibition area!

Any Questions?
For more information about Vector and our products please visit

www.vector.com

Author: Neuffer, Jochen
Vector Germany