

Continuous Integration
with Silver at
Volvo Cars Corporation

Vehicle Energy & Mode Control

Johannes Foufas





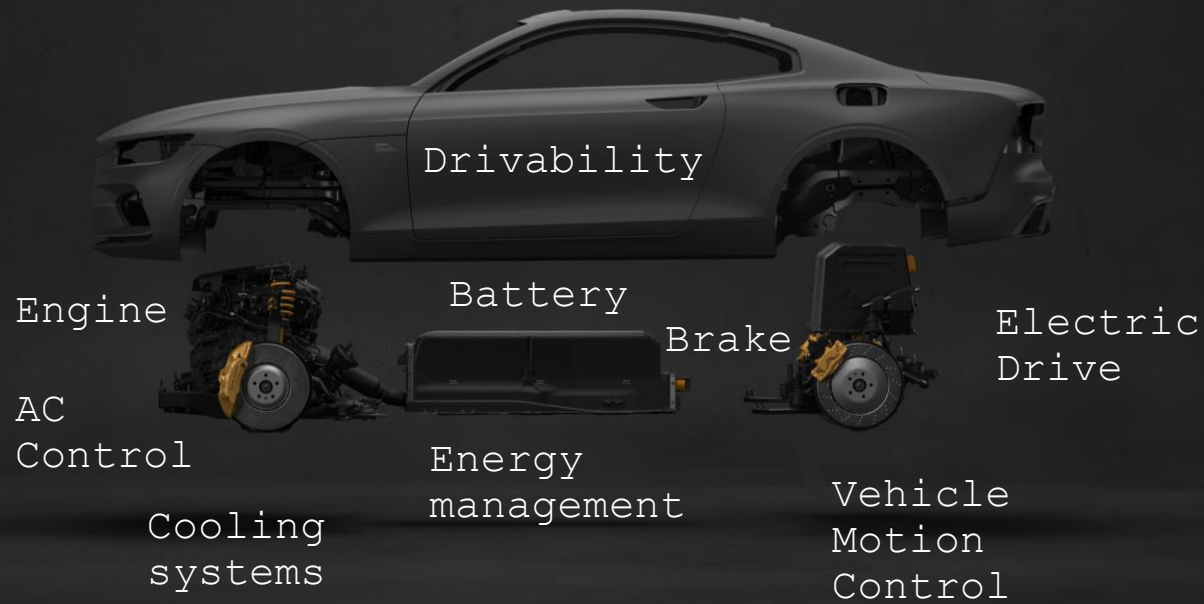
Polestar 1

—
VP Driver experience test

Usage of Silver is growing

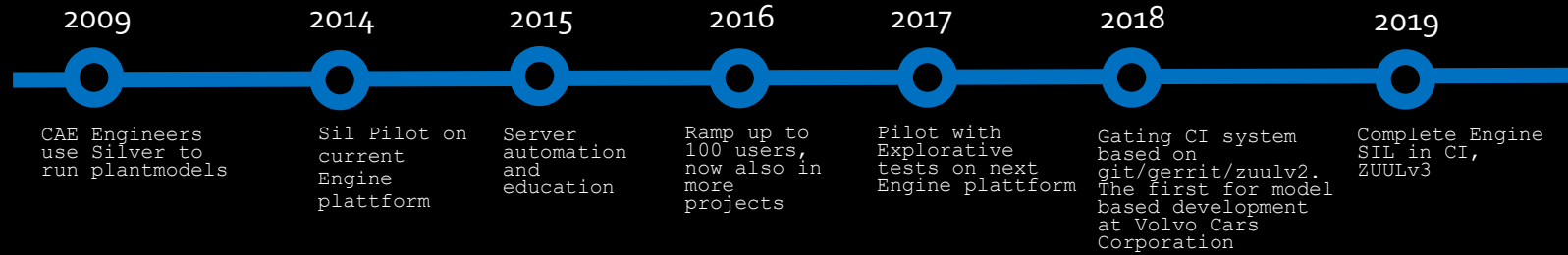


Usage of Silver is growing



Time line

An organic growth of SW components and new users





<https://zuul-ci.org/>

Stop Merging Broken Code

Tested at scale

Zuul powers some of the largest Open Source development efforts,
and we contribute!



ANSIBLE

Project Gating

Keep your builds evergreen by automatically merging changes only if they pass tests.

CI/CD with Ansible

Use the same Ansible playbooks to deploy your system and run your tests.

Cross-Project Dependencies

Easily test changes to multiple systems together before landing a single patch.

Zuul tests cross-project changes in parallel.

This is essential for projects with many ECU's

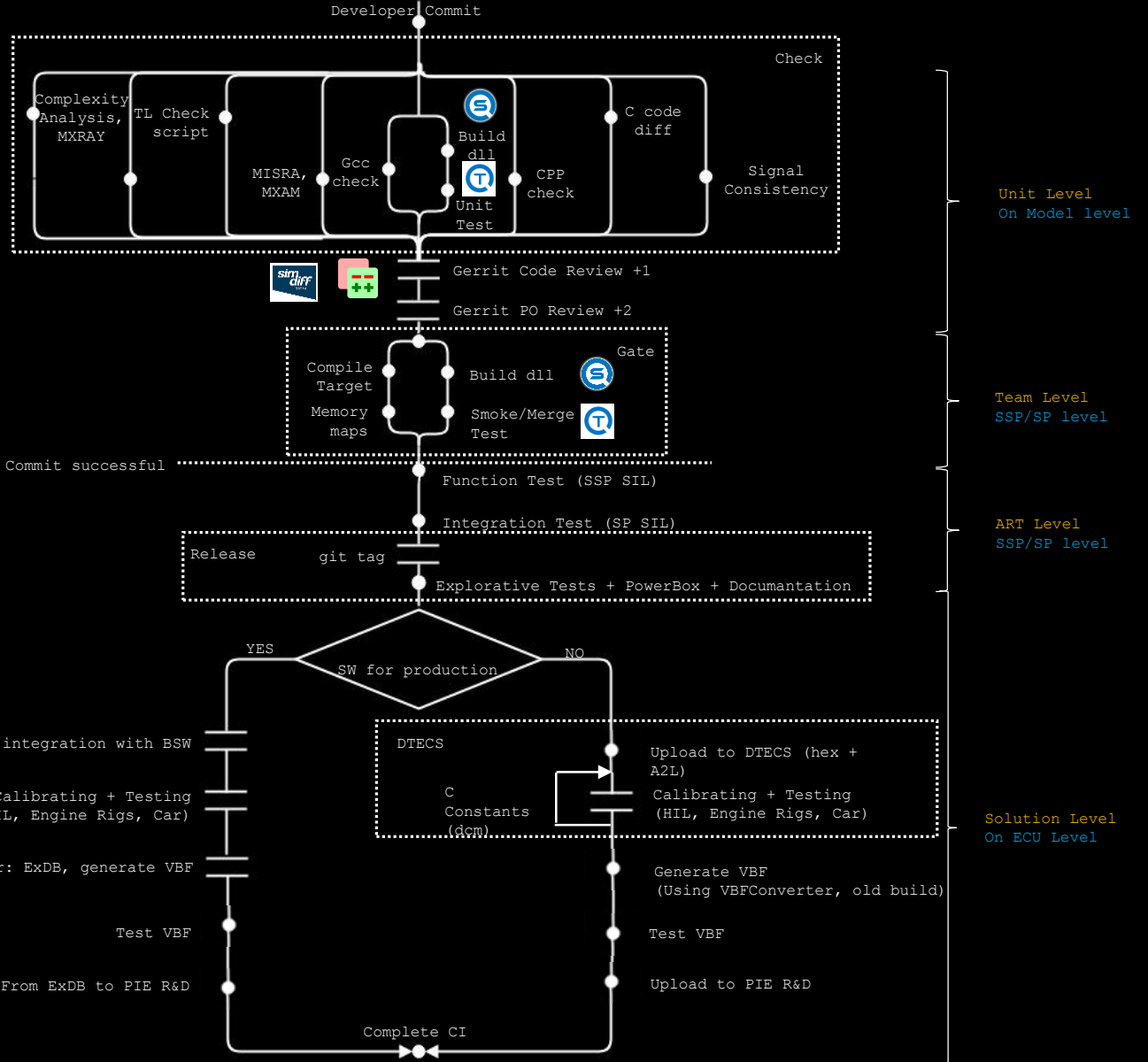
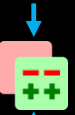
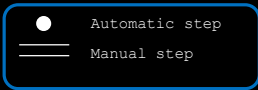
Gating

```
/ Every change proposed for a repository \  
\ is tested before it merges. \  
-----
```



gate

0



check

1

Newly uploaded patchsets enter this pipeline to receive an initial
+/-1 Verified vote You can retrigger this check by
commenting "recheck" in Gerrit.

pt_pcc	5 min
15185,2	0 min
pybuild_diff	<input type="checkbox"/>
Pybuild_CppCheck	<input type="checkbox"/>
pybuild_gcc_check	<input type="checkbox"/>
pybuild_checksript	<input type="checkbox"/>
UnitTests_shared_functions	success
_UnitTests_pybuild	success
_UnitTests_pybuild	success
UnitTests_pybuild	success
UnitTests_pybuild	success
Signal_consistency	<input type="checkbox"/>
SIMDIFF	<input type="checkbox"/>
MXRAY_Report	<input type="checkbox"/>
MXAM_Report	<input type="checkbox"/>
CI_of_CI	success

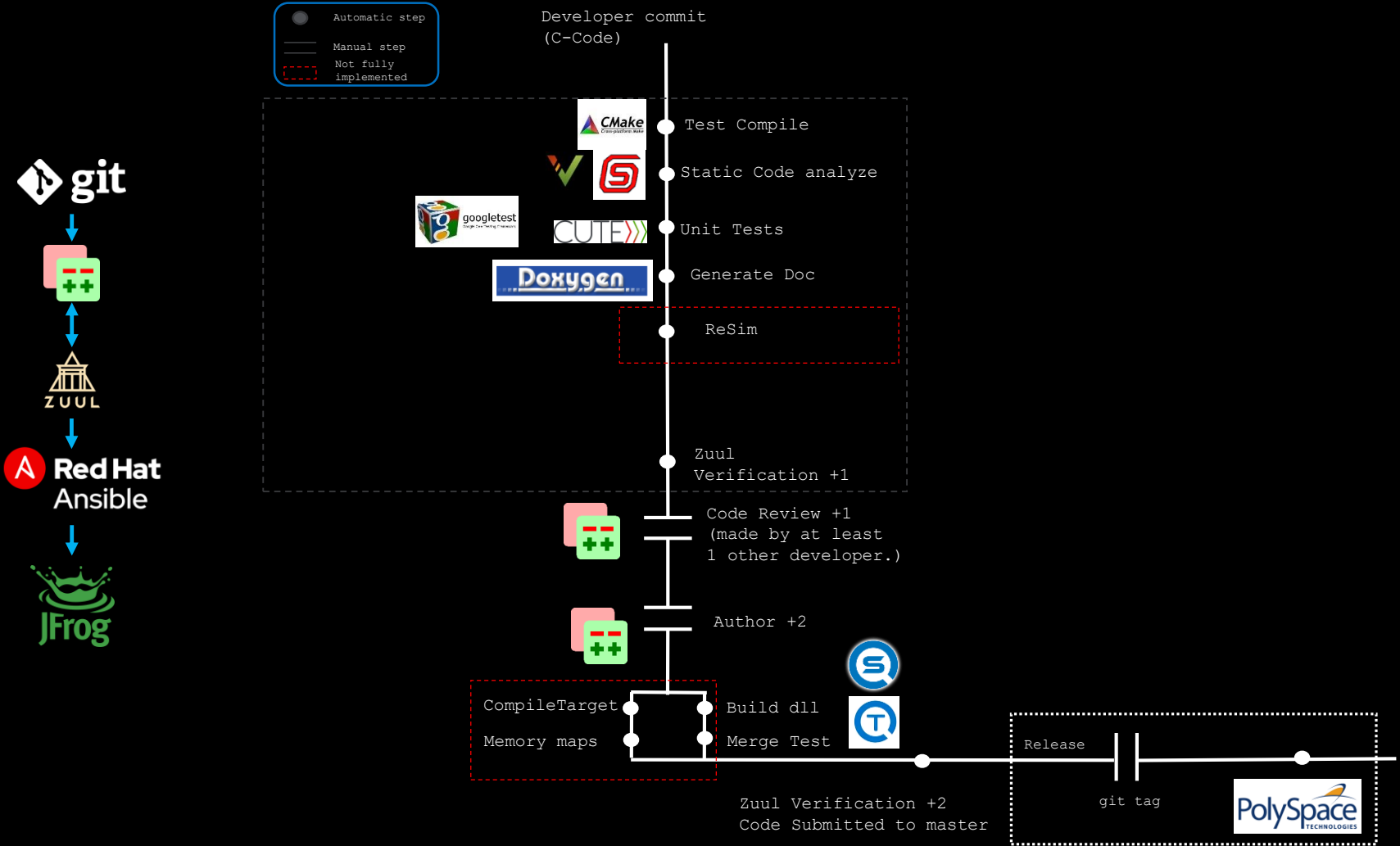
gate

1

Changes that have been approved by core developers are
enqueued in order in this pipeline. Retriggered by writing "regate"
in Gerrit.

pt_pcc	0 min
15185,2	0 min
023_Compile_pybuild	<input type="checkbox"/>
034_ECM_SIL_DLL	<input type="checkbox"/>
035_ECM_SIL_MergeTests	queued
034_ECM_SIL_DLL	<input type="checkbox"/>
035_ECM_SIL_MergeTests	queued
Pybuild_MergeTests	<input type="checkbox"/>
023_Compile_pybuild	<input type="checkbox"/>
Pybuild_MergeTests	<input type="checkbox"/>
23_Compile_pybuild	<input type="checkbox"/>
Pybuild_MergeTests	<input type="checkbox"/>
23_Compile_pybuild	<input type="checkbox"/>
Pybuild_MergeTests	queued
023_Compile_pybuild	<input type="checkbox"/>
Pybuild_MergeTests	queued

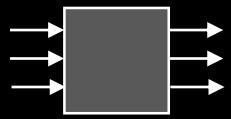




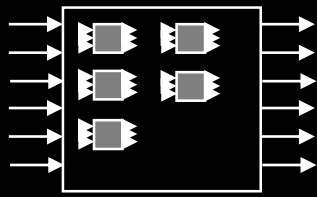
SIL Test levels



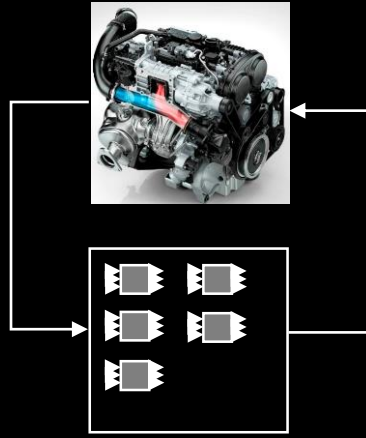
- Unit-, Module- and ECM-level tests
- ECM includes supplier SW
- With or without plant models
- All SIL tests - from open-loop unit tests to closed-loop ECM tests are using the same toolchain



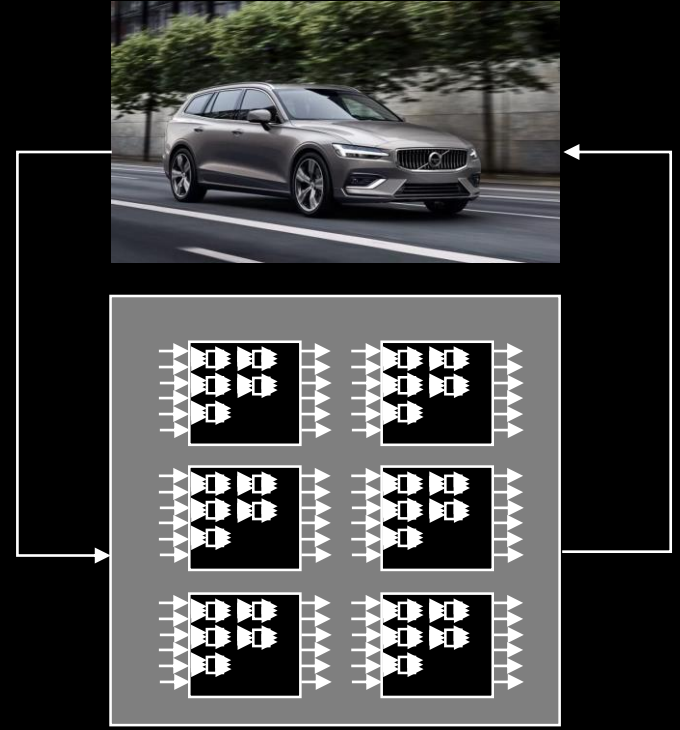
Unit SIL test
Open-loop



Module SIL test
Open-loop



Module SIL test
Closed-loop



ECM SIL test
Closed-loop

What makes it work?

- * Non democratic
 - small core team decides, backed up by management
- * Pair debugging with developers
- * Distribute to developers machines
- * Gating, don't merge broken code!
 - The heart is SIL merge tests, component tests with sensor models
 - Speculative merge, all changes tested sequential in parallel
- * Incremental changes
 - added jobs, tests, checks, required levels to pass
 - Features added over time, like custom builds/jobs
 - Performance improved over time, build avoidance, 38min average

ZUUL v2.6 problems

* Updates to zuul or jenkins needs compatible to all branches.

There is no Jenkins only Zuul v3

- * Ansible replaces Jenkins job execution.
- * Nodepool replaces Jenkins node management.
- * Everything is yaml!

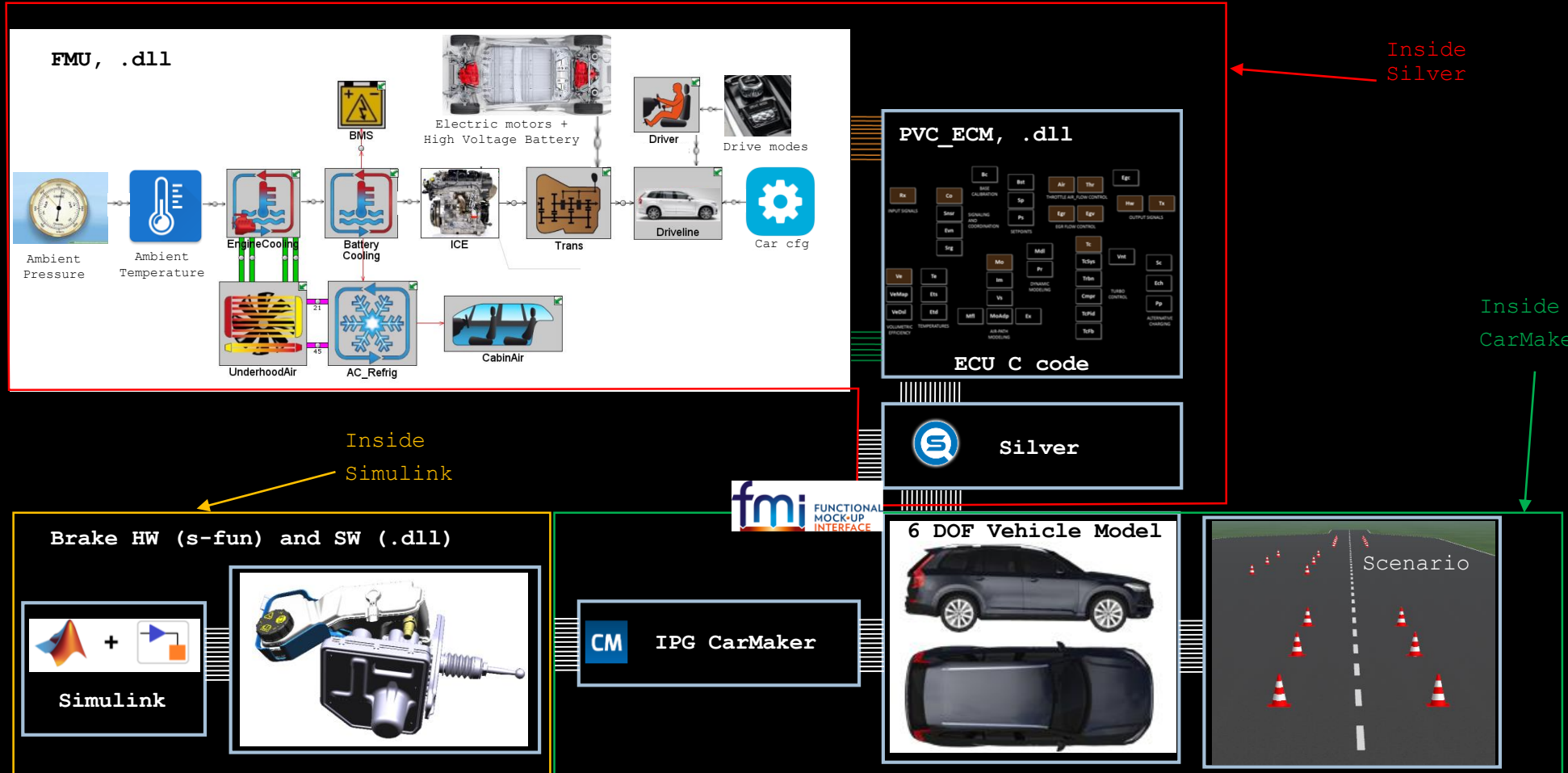
```
          D---E master
         /
        A---B---C---F origin/master
```

It puts the CI configuration in the repo

- * Configuration is synchronized with build items.
 - No need to think about backwards compatibility.
- * Infrastructure can be tested together with the rest of the project.
 - No unicorns in the server hall!
 - All jobs are reproducible!

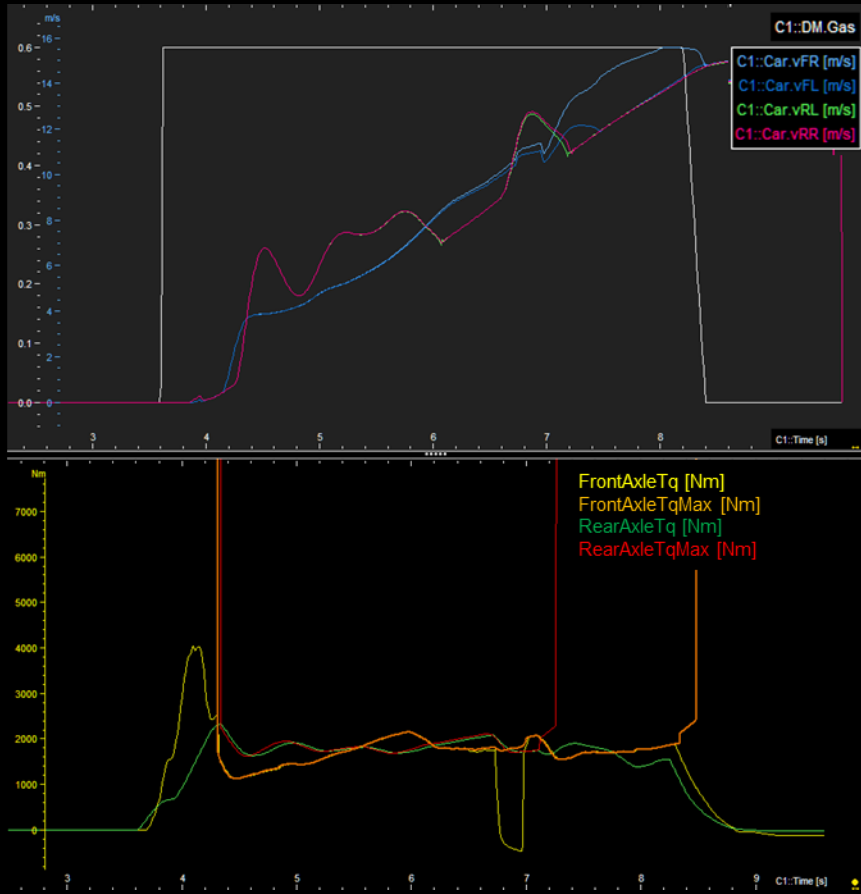
Silver – CarMaker, Vehicle Motion POC

Proof of concept, by Alejandro Gonzalez and his team.

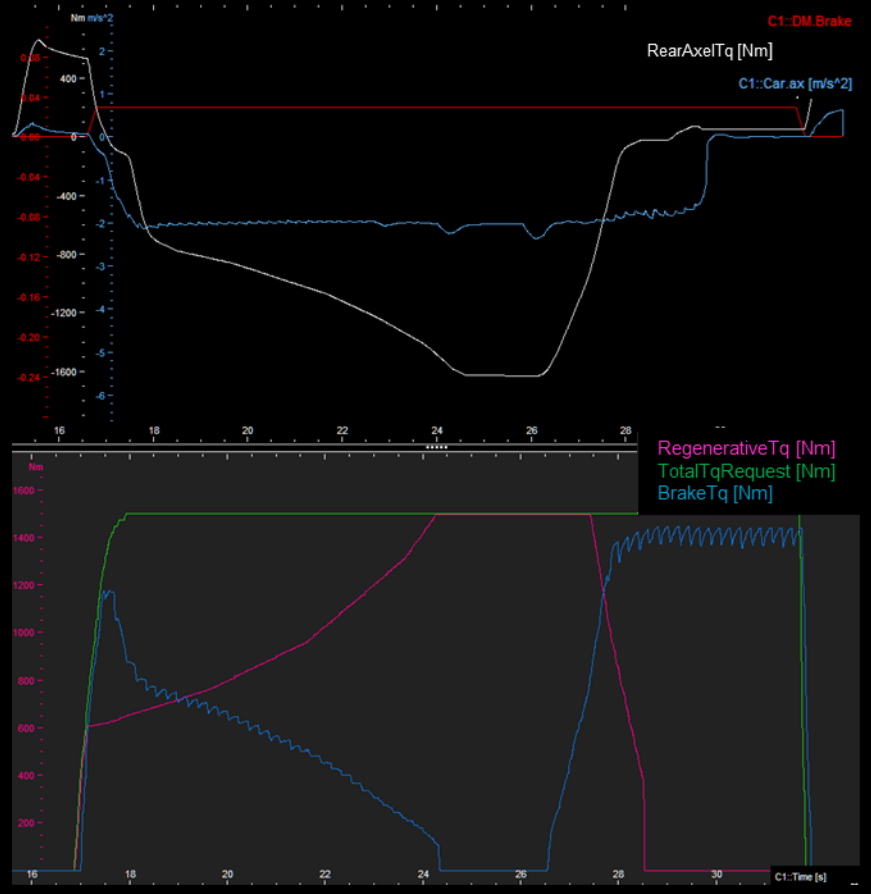


Silver – CarMaker, Vehicle Motion POC

Traction Control



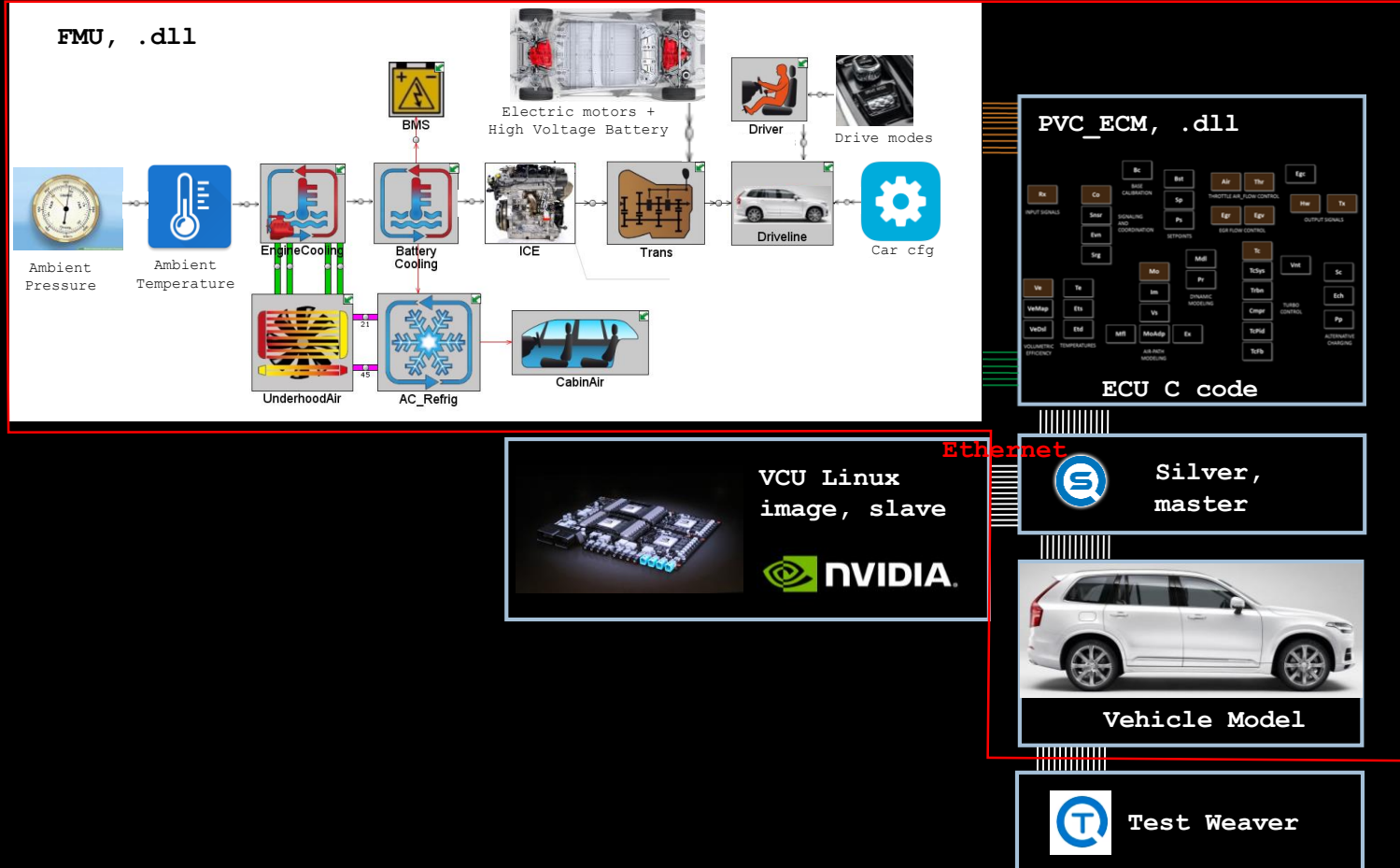
Brake Blending



Current development, Domain SII



Volvo Cars Corporation & Synopsis Qtronic cooperates and builds a setup where Silver is master to a Linux image slave of a VCU. The core computer is based on NVIDIA's DRIVE AGX Xavier technology and will allow Volvo Cars to implement an advanced computing platform for its new cars on the forthcoming Scalable Product Architecture 2 (SPA 2) vehicle platform.



Inside Silver

Ethernet