Introduction to PopcornSAR Adaptive AUTOSAR (R20-11) Tool chain

March. 2022

PopcornSAR

Index

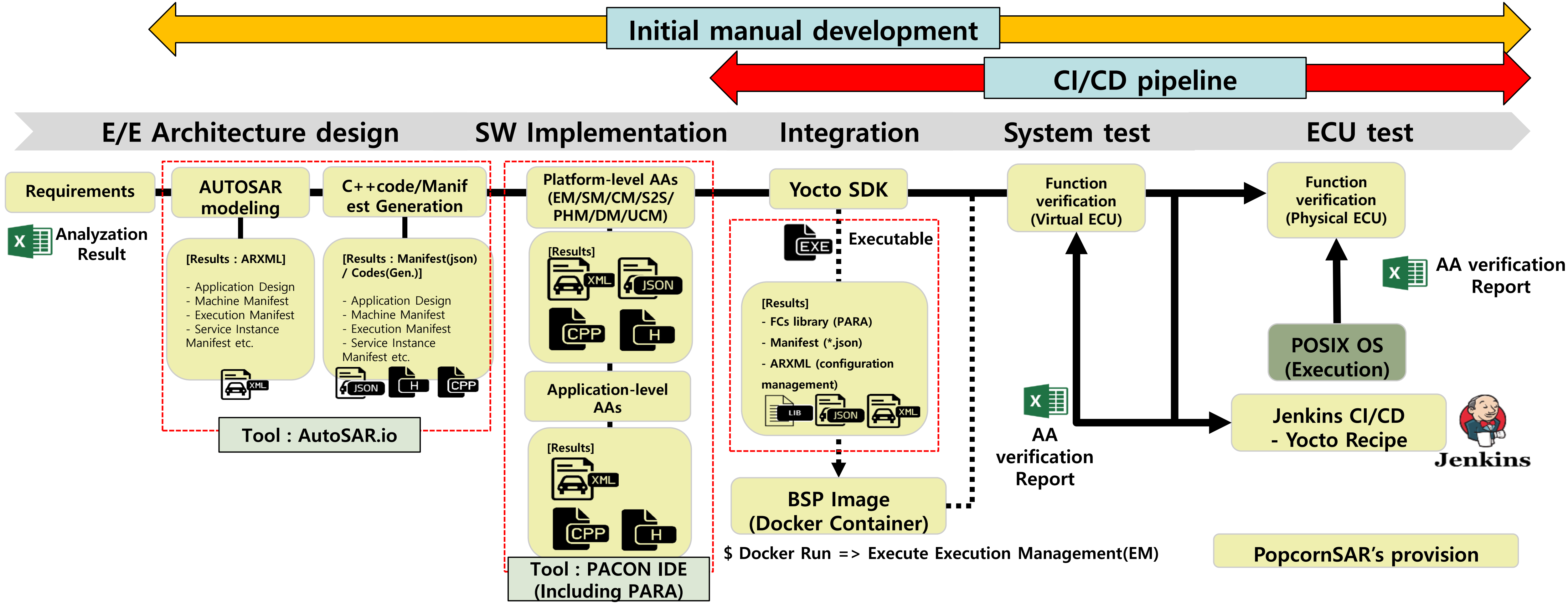
1. Automotive Software Development Process with Adaptive AUTOSAR (R20-11)
2. Overview of PopcornSAR Adaptive AUTOSAR (R20-11) tool chain
3. Product introduction: PARA
4. Product introduction: PACON IDE
5. Product introduction: Virtual ECU
6. Example of PopcornSAR AP development environment
7. PopcornSAR AP tool roadmap(~2023)

(Appendix 1) Tool demonstrations

(Appendix 2) ISO26262 Certificate of PopcornSAR Adaptive platform Authoring tool

1. Automotive Software Development Process with Adaptive AUTOSAR (R20-11)

1. PopcornSAR provides automotive software development tool chain for Adaptive AUTOSAR Applications(AA).
   - Supporting the whole development process of Adaptive Applications/ECUs for OEM/Tier1.
2. Able to automate the whole development process based on CI/CD after the initial manual development.
3. Yocto SDK is used for Adaptive application(AA) development after the SOP.



Initial manual development

CI/CD pipeline

E/E Architecture design | SW Implementation | Integration | System test | ECU test

Requirements

Analyzation Result

AUTOSAR modeling

[Results : ARXML]
- Application Design
- Machine Manifest
- Execution Manifest
- Service Instance Manifest etc.

C++code/Manifest Generation

[Results : Manifest(json) / Codes(Gen.)]
- Application Design
- Machine Manifest
- Execution Manifest
- Service Instance Manifest etc.

Tool : AutoSAR.io

Platform-level AAs (EM/SM/CM/S2S/PHM/DM/UCM)

[Results]

Application-level AAs

[Results]

Tool : PACON IDE (Including PARA)

Yocto SDK

Executable

[Results]
- FCs library (PARA)
- Manifest (*.json)
- ARXML (configuration management)

BSP Image (Docker Container)

$ Docker Run => Execute Execution Management(EM)

Function verification (Virtual ECU)

AA verification Report

Function verification (Physical ECU)

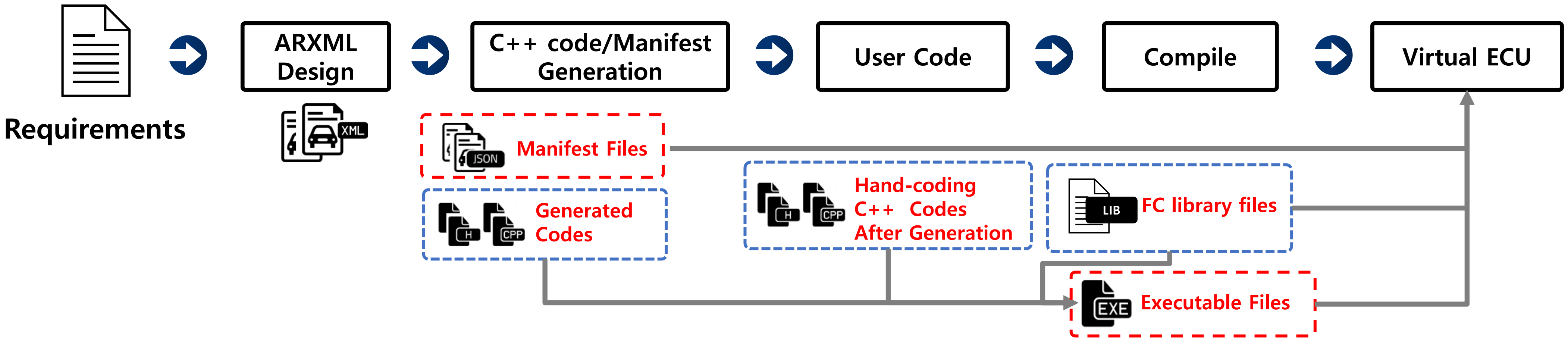
AA verification Report

POSIX OS (Execution)

Jenkins CI/CD
- Yocto Recipe

PopcornSAR's provision

1. Automotive Software Development Process with Adaptive AUTOSAR (R20-11)



Requirements → ARXML Design → C++ code/Manifest Generation → User Code → Compile → Virtual ECU

Manifest Files

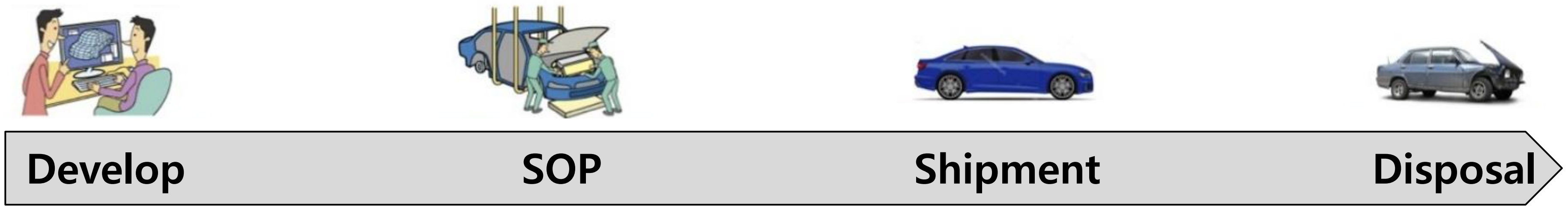
Generated Codes

Hand-coding C++ Codes After Generation

FC library files

Executable Files

- **After testing on virtual ECU, developers need to conduct system test on the physical ECU.**



Develop **SOP** **Shipment** **Disposal**

AP version (R20-11)

Upgrade AP version through OTA (R21-11)

- **Vehicle functions and cyber security can be updated through OTA after SOP.**

2. Overview of PopcornSAR Adaptive AUTOSAR (R20-11) tool chain

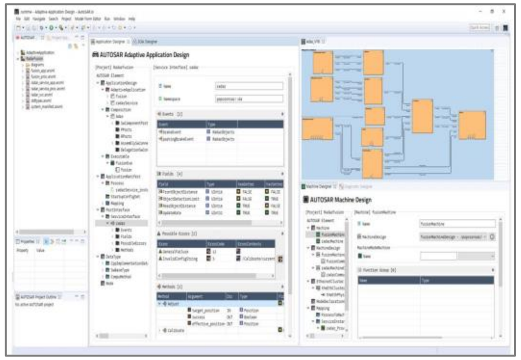
- A development platform with tools installed in PC

  ① SDK of Adaptive Platform(AP) Functional Cluster(FC) or Vehicle APIs (No dependency on middleware vendor).

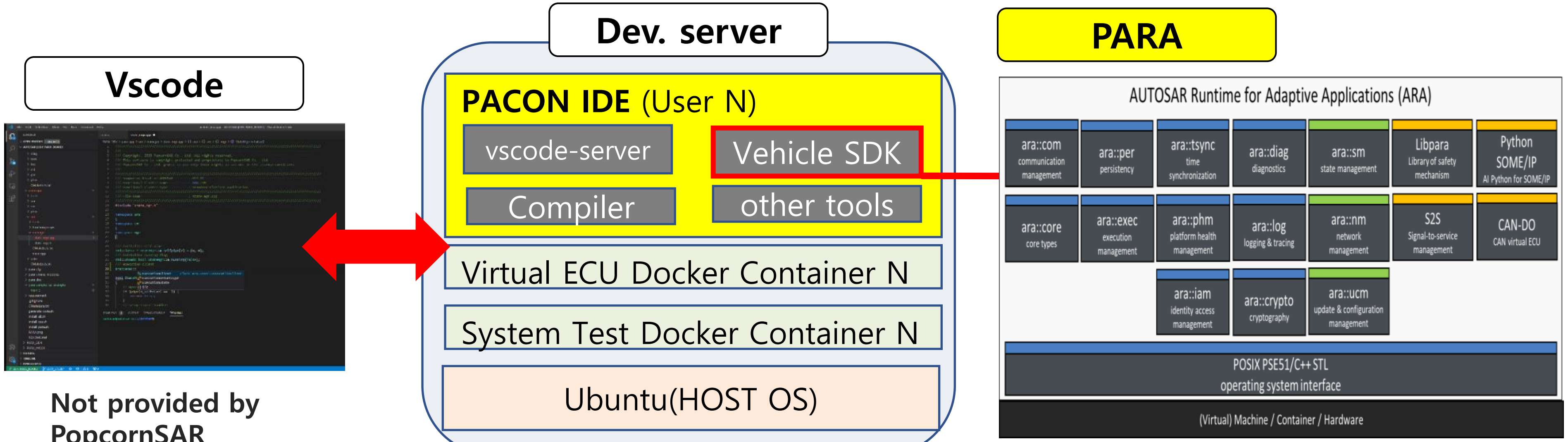
  ② Providing virtual ECU (simulation & test), supporting programing languages like C＋＋.

  ③ Coding Rule check, API autocomplete, etc.

AutoSAR.io



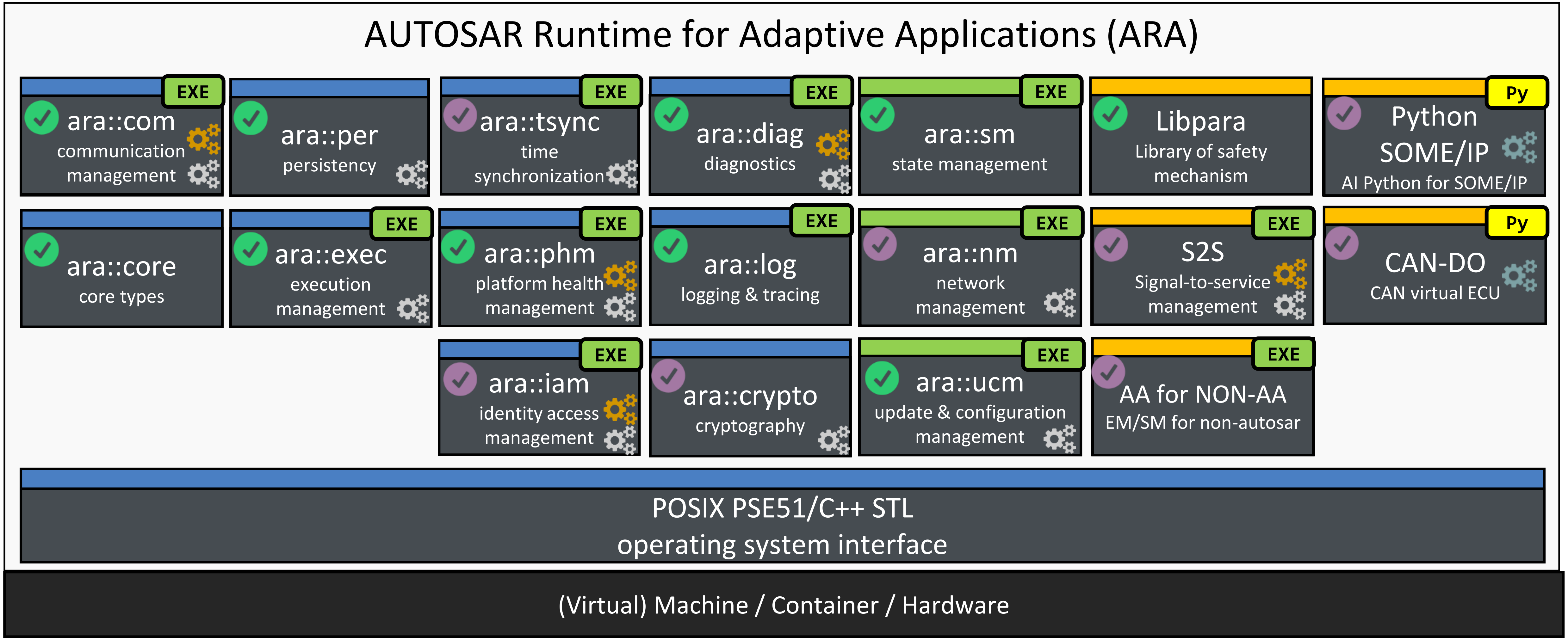
ARXML modeling tool
(Supports AP & CP)


Vscode
Dev. server
PACON IDE (User N)
vscode-server
Vehicle SDK
Compiler
other tools
Virtual ECU Docker Container N
System Test Docker Container N
Ubuntu(HOST OS)
PARA
AUTOSAR Runtime for Adaptive Applications (ARA)
ara::com communication management
ara::per persistency
ara::tsync time synchronization
ara::diag diagnostics
ara::sm state management
Libpara Library of safety mechanism
Python SOME/IP AI Python for SOME/IP
ara::core core types
ara::exec execution management
ara::phm platform health management
ara::log logging & tracing
ara::nm network management
S2S Signal-to-service management
CAN-DO CAN virtual ECU
ara::iam identity access management
ara::crypto cryptography
ara::ucm update & configuration management
POSIX PSE51/C++ STL
operating system interface
(Virtual) Machine / Container / Hardware

Not provided by PopcornSAR

Adaptive Platform Functional Cluster(FC) & Platform Level Adaptive Applications

3. Product Introduction : **PARA** - FCs list (R20-11) To-be (before 2023)



AUTOSAR Runtime for Adaptive Applications (ARA)

EXE ✓ ara::com communication management

✓ ara::per persistency

EXE ✓ ara::tsync time synchronization

EXE ✓ ara::diag diagnostics

EXE ✓ ara::sm state management

✓ Libpara Library of safety mechanism

Py ✓ Python SOME/IP AI Python for SOME/IP

✓ ara::core core types

EXE ✓ ara::exec execution management

EXE ✓ ara::phm platform health management

EXE ✓ ara::log logging & tracing

EXE ✓ ara::nm network management

EXE ✓ S2S Signal-to-service management

Py ✓ CAN-DO CAN virtual ECU

EXE ✓ ara::iam identity access management

✓ ara::crypto cryptography

EXE ✓ ara::ucm update & configuration management

EXE ✓ AA for NON-AA EM/SM for non-autosar

POSIX PSE51/C++ STL
operating system interface

(Virtual) Machine / Container / Hardware

✓ Available now

✓ Available before 2023

⚙ C++ Source Code Generator

⚙ Manifest Generator

⚙ Specific Generator

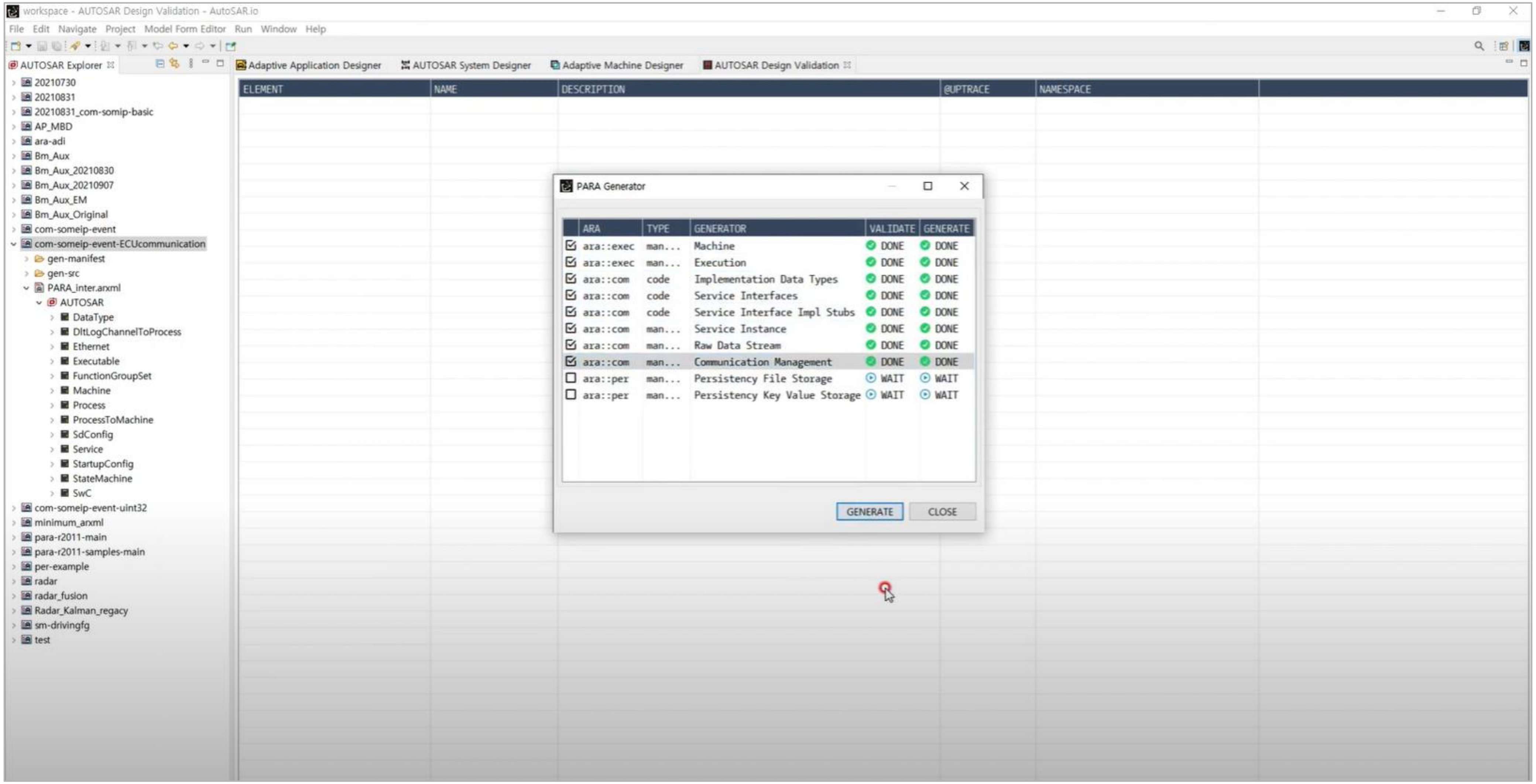
EXE Platform Level Application

🟨 PopcornSAR Specific

3. Product introduction: PARA - Features

1. **PARA enables Adaptive Application tests without Execution Management(EM).**
   - According to AUTOSAR standard, EM is essential. However, PARA users can develop/test adaptive applications without EM, increasing development efficiency.
2. **If no problem is found in ARXML Validation, no more parameter is needed for PopcornSAR's generators, and the generation will be done automatically.**
   - AutoSAR.io includes generators of FC, so it automatically analyzes Adaptive Applications and Machine, and generates related source codes and Manifest files.
3. **PopcornSAR can migrate results(C++) from Matlab/Simulink 2021a(R19 - 11) or AP M odel Based Design(MBD), Legacy MBD into Adaptive applications, and related guidelines are also available.**
   - It's necessary to use Matlab/Simulink to develop Adaptive Applications for control.
4. **PARA supports Python's SOME/IP for AI development.**

3. Product introduction: PARA - Features

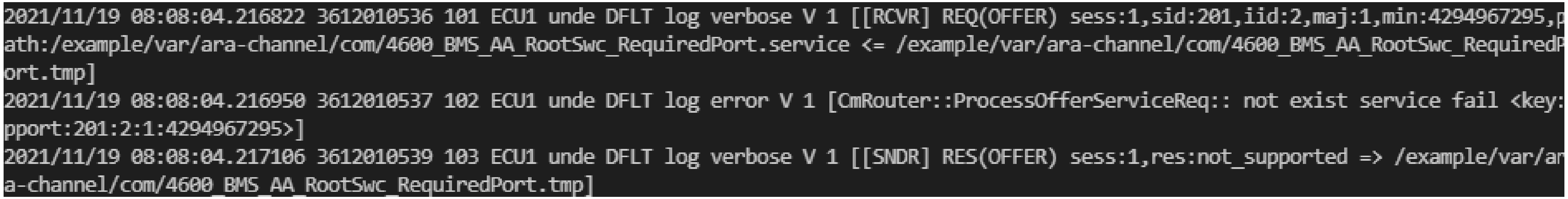


	ARA	TYPE	GENERATOR	VALIDATE	GENERATE
☑	ara::exec	man...	Machine	✓ DONE	✓ DONE
☑	ara::exec	man...	Execution	✓ DONE	✓ DONE
☑	ara::com	code	Implementation Data Types	✓ DONE	✓ DONE
☑	ara::com	code	Service Interfaces	✓ DONE	✓ DONE
☑	ara::com	code	Service Interface Impl Stubs	✓ DONE	✓ DONE
☑	ara::com	man...	Service Instance	✓ DONE	✓ DONE
☑	ara::com	man...	Raw Data Stream	✓ DONE	✓ DONE
☑	ara::com	man...	Communication Management	✓ DONE	✓ DONE
□	ara::per	man...	Persistency File Storage	○ WAIT	○ WAIT
□	ara::per	man...	Persistency Key Value Storage	○ WAIT	○ WAIT

<Generation finishes in average of 1 minute by PARA in AutoSAR.io>

3. Product introduction: PARA – Safety Mechanism for ISO26262

- PopcornSAR provides special safety mechanism(libpara) which is developed by PopcornSAR own. It can detect faults in execution level when AA is running, such as:
  1) Human errors in using ARA API when implementing Adaptive Applications,
  2) Human errors in using POSIX OS.



```
2021/11/19 08:08:04.216822 3612010536 101 ECU1 unde DFLT log verbose V 1 [[RCVR] REQ(OFFER) sess:1,sid:201,iid:2,maj:1,min:4294967295,p
ath:/example/var/ara-channel/com/4600_BMS_AA_RootSwc_RequiredPort.service <= /example/var/ara-channel/com/4600_BMS_AA_RootSwc_RequiredP
ort.tmp]
2021/11/19 08:08:04.216950 3612010537 102 ECU1 unde DFLT log error V 1 [CmRouter::ProcessOfferServiceReq:: not exist service fail <key:
pport:201:2:1:4294967295>]
2021/11/19 08:08:04.217106 3612010539 103 ECU1 unde DFLT log verbose V 1 [[SNDR] RES(OFFER) sess:1,res:not_supported => /example/var/ar
a-channel/com/4600_BMS_AA_RootSwc_RequiredPort.tmp]
```

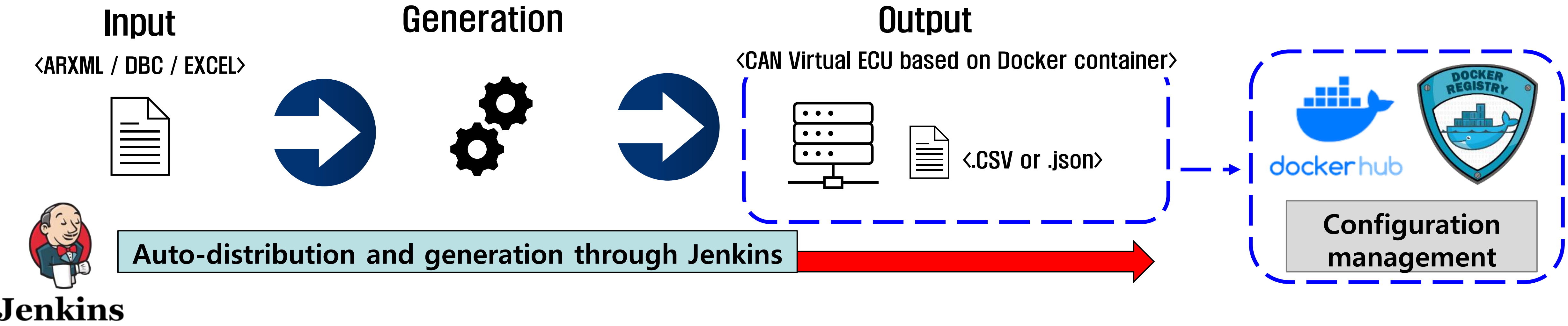
<ex1 Human error of ara API : SOME/IP Error fault detection when AA runs: someip OfferService API>

```
2022/02/09 03:39:18.645278 102307244 001 ECU1 CM-- DFLT log error V 1 [InitByManifest:: ManifestParser::InitByManifest::
HasParserError fail <0> (/home/popcornsar/para-r2011-main/para-api/com/internal/database/database.cpp #63)] 5terminate
called after throwing an instance of 'std::runtime_error' 6 what(): InitByManifest:: ManifestParser::InitByManifest::
HasParserError fail <0> (/home/popcornsar/para-r2011-main/para-api/com/internal/database/database.cpp #63
```

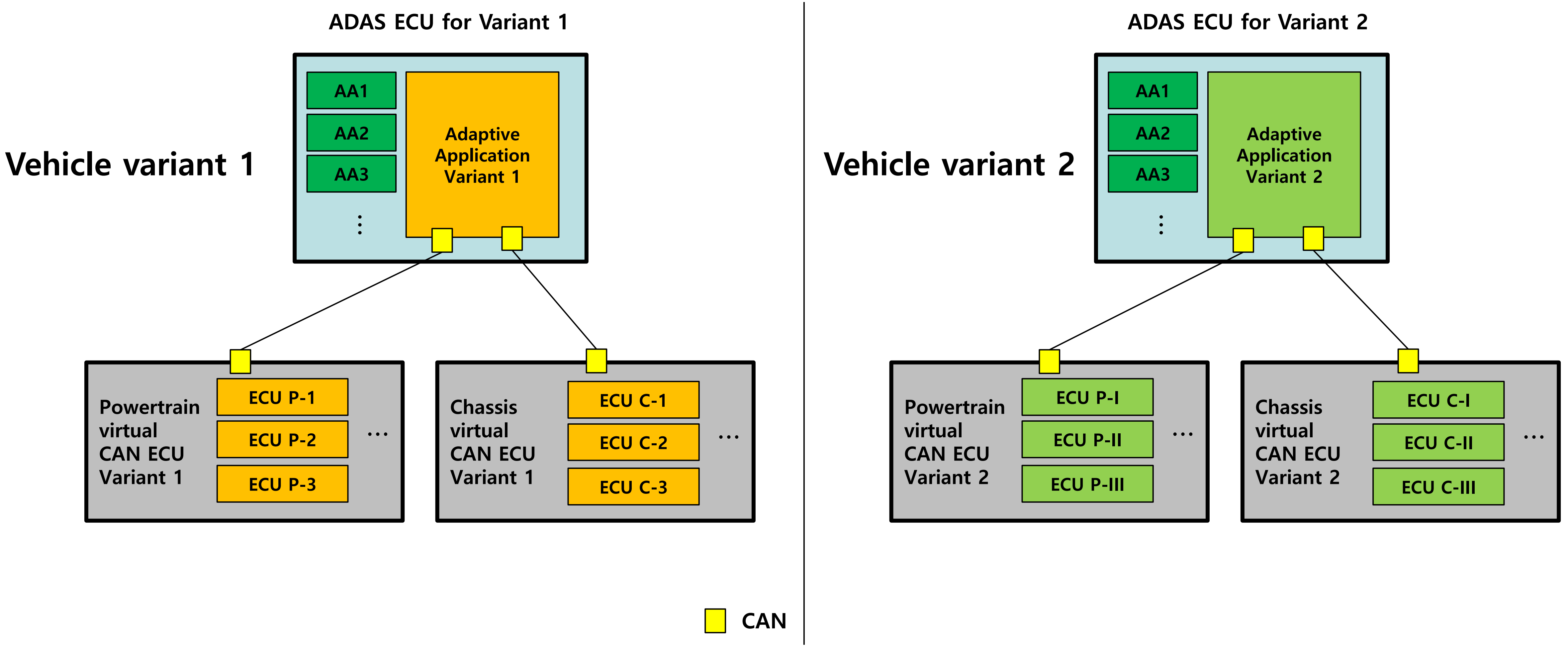
<ex2 Human error of POSIX OS : dummy files created when using vi editor of POSIX OS in target board>

3. Product introduction: PARA – CAN-DO(based on Docker container)

- CAN-DO can generate CAN virtual ECU which is based on docker container with ARXML/DBC/EXCEL.
- Able to fast generate multiple CAN virtual ECUs for testing ECUs.
- Able to manage CAN virtual ECU's configurations through Dockerhub, and able to automate the virtual ECU generation process through Jenkins.
- No limitation of AUTOSAR versions.



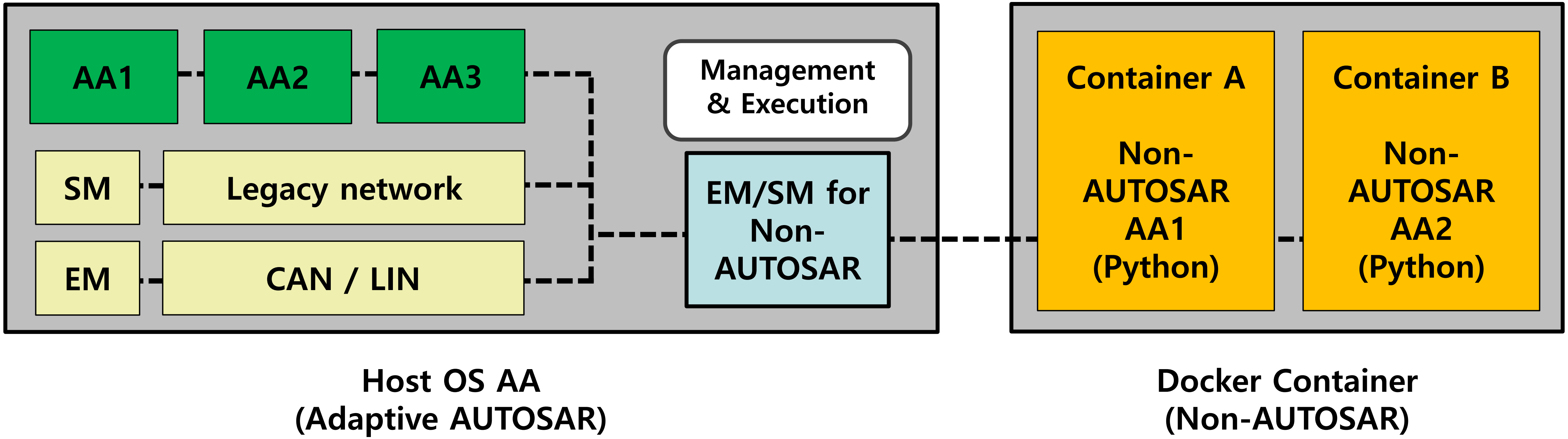
3. Product introduction: PARA – Example use case of CAN-DO



ADAS ECU for Variant 1
Vehicle variant 1
AA1
AA2
AA3
⋮
Adaptive Application Variant 1
Powertrain virtual CAN ECU Variant 1
ECU P-1
ECU P-2
ECU P-3
…
Chassis virtual CAN ECU Variant 1
ECU C-1
ECU C-2
ECU C-3
…
🟨 CAN
ADAS ECU for Variant 2
Vehicle variant 2
AA1
AA2
AA3
⋮
Adaptive Application Variant 2
Powertrain virtual CAN ECU Variant 2
ECU P-I
ECU P-II
ECU P-III
…
Chassis virtual CAN ECU Variant 2
ECU C-I
ECU C-II
ECU C-III
…

- Virtual CAN ECUs can be generated by CAN-DO automatically.
- Configurations can be saved & managed by Dockerhub/Docker registry.

3. Product introduction: PARA – AA for Non-AUTOSAR Applications execution



- EM might not recognize non-AUTOSAR Apps (docker, legacy network app, Python etc).
- PopcornSAR provides special AA as a role of EM/SM to execute Non-AUTOSAR Apps.

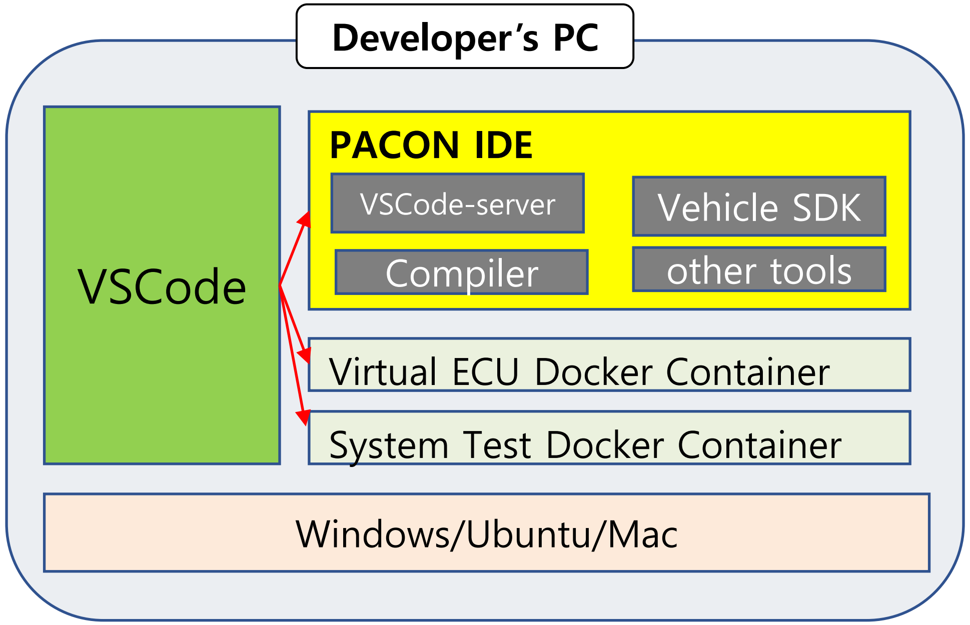
4. Product introduction: PACON IDE – Features

1. **PACON IDE(=Docker Container) can be customized according to customer's needs.**
   - Separate IDEs for different variants.
   - Customer's IDEs can be redistributed & managed through Jenkins CI/CD.
2. **Has no dependency on Adaptive Platform Vendor.**
3. **Code developer can use Vscode to connect with PACON IDE.**
   - Docker Container has essential extensions for using PARA & Vscode.
   - External network is not needed.
4. **Wireshark is provided even if it's not installed in developer's PC.**
5. **Can be distributed through Jenkins CI/CD.**
6. **Provides additional features for efficient development.**
   - Auto complete function for ARA API & open source API.
   - Real-time code rule check.
   - Debug.
   - Various programming languages (Python etc.,).

4. Product introduction: PACON IDE – Two variations of PACON IDE

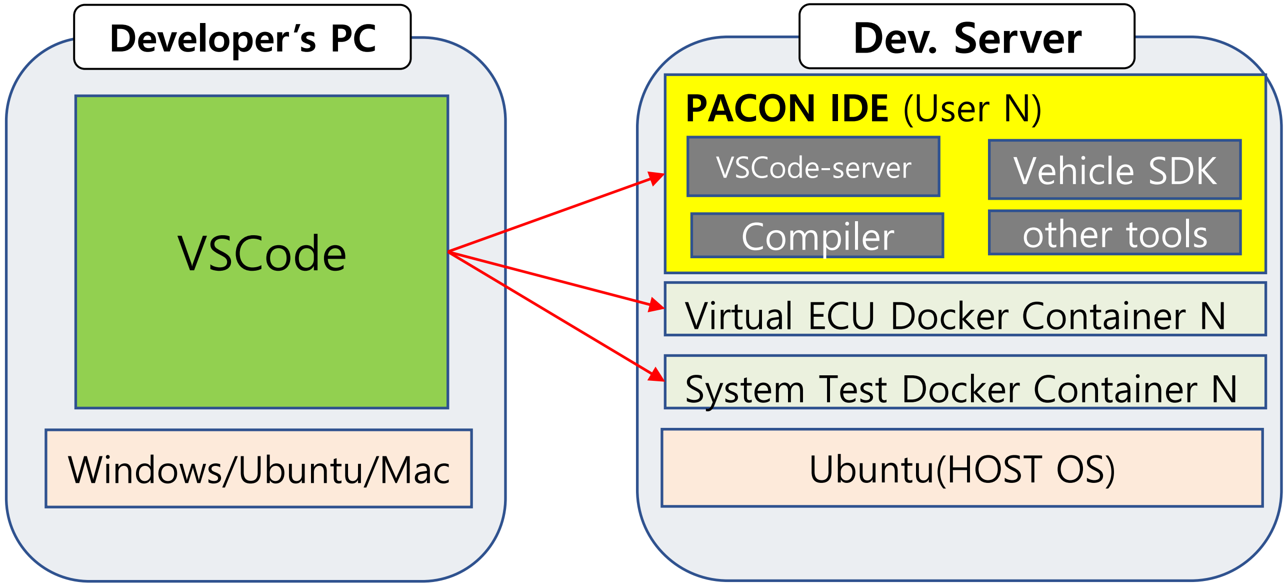
① Develop within developer's PC

(Without external network)



② With connection to development server

(Increase convenience & efficiency based on remote co-working environment)



docker container	: Distributed through Dockerhub

4. Product introduction: PACON IDE – Customization according to requirements

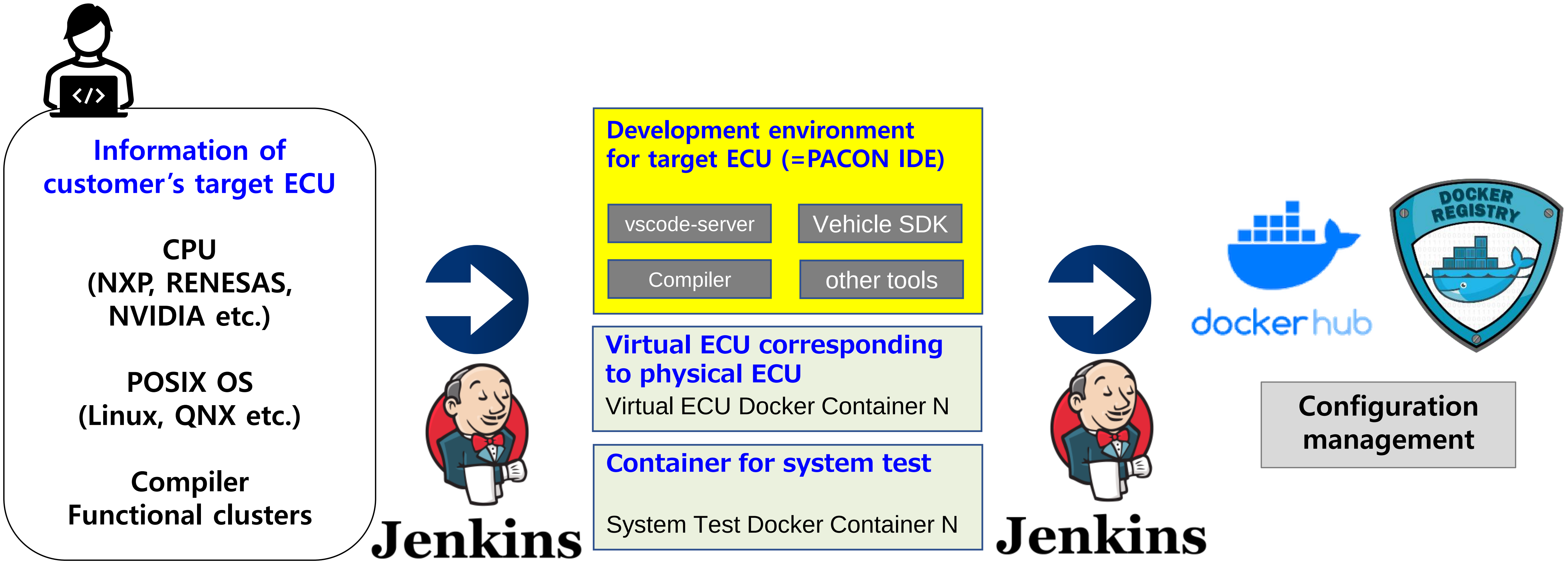
Variant	CPU	OS	
Variant 1	NVIDIA	Linux (5.0)	**Variant1:1.0.0**
	NVIDIA	Linux (5.15)	**Variant1:1.1.0**
Variant 2	Intel	QNX	**Variant2_intel:1.0.0**
	NXP	QNX	**Variant2_nxp:1.0.0**
Variant 3	R-car	Android	**Variant3:1.0.0**

- Separate IDEs for different variants.
- Customer IDE can be redistributed & managed through Jenkins CI/CD.
- IDE must be redistributed according to updates of OS.

Jenkins

Auto-distribution
through Jenkins

4. Product introduction: PACON IDE - IDE and Virtual ECU creation process



Information of customer's target ECU

CPU
(NXP, RENESAS, NVIDIA etc.)

POSIX OS
(Linux, QNX etc.)

Compiler
Functional clusters

Development environment for target ECU (=PACON IDE)

vscode-server

Vehicle SDK

Compiler

other tools

Virtual ECU corresponding to physical ECU

Virtual ECU Docker Container N

Container for system test

System Test Docker Container N

Configuration management

5. Product introduction: Virtual ECU – Features (1)

1. **PopcornSAR's virtual ECU can be customized according to customer's requirements.**
   - Separate virtual ECUs for different variants.
   - Customer's virtual ECUs can be redistributed & managed through Jenkins CI/CD.
2. **Virtual ECU has no dependency on Adaptive Platform Vendor.**
3. **PopcornSAR's Virtual ECU has lighter footprint, inducing efficient development with less compatibility issues than QEMU.**
   - PopcornSAR provides virtual ECU(docker container) which corresponds to customer's physical ECU requirements(POSIX OS etc.).
   - After being tested on Docker Container, adaptive applications can be copied directly on target ECU.

5. Product introduction: Virtual ECU – Features (2)

4. **Developers can generate virtual ECUs through PACON IDE.**
5. **Developers can develop multiple ECUs in docker container, manage their configurations through Dockerhub or Docker registry, and share their results with other colleagues.**
6. **Multiple virtual ECUs can be generated by Jenkins CI/CD for adaptive application tests.**
7. **PACON IDE provides additional docker containers for virtual ECUs to carry out other system tests.**
   - Usually, it's difficult to add system testing software in virtual ECU. (Ex : Installing tshark in virtual ECU ARM)
   - PopcornSAR can generate additional docker containers for system testing software.

5. Product introduction: Virtual ECU – Customization according to requirements

- IDE and virtual ECU for the same variant are managed separately.

Variant	CPU	OS
Variant 1	NVIDIA	Linux (5.0)
	NVIDIA	Linux (5.15)
Variant 2	Intel	QNX
	NXP	QNX
Variant 3	R-car	Android

vECU_Variant1:1.0.0

vECU_Variant1:1.1.0

vECU_Variant2_intel:1.0.0

vECU_Variant2_nxp:1.0.0

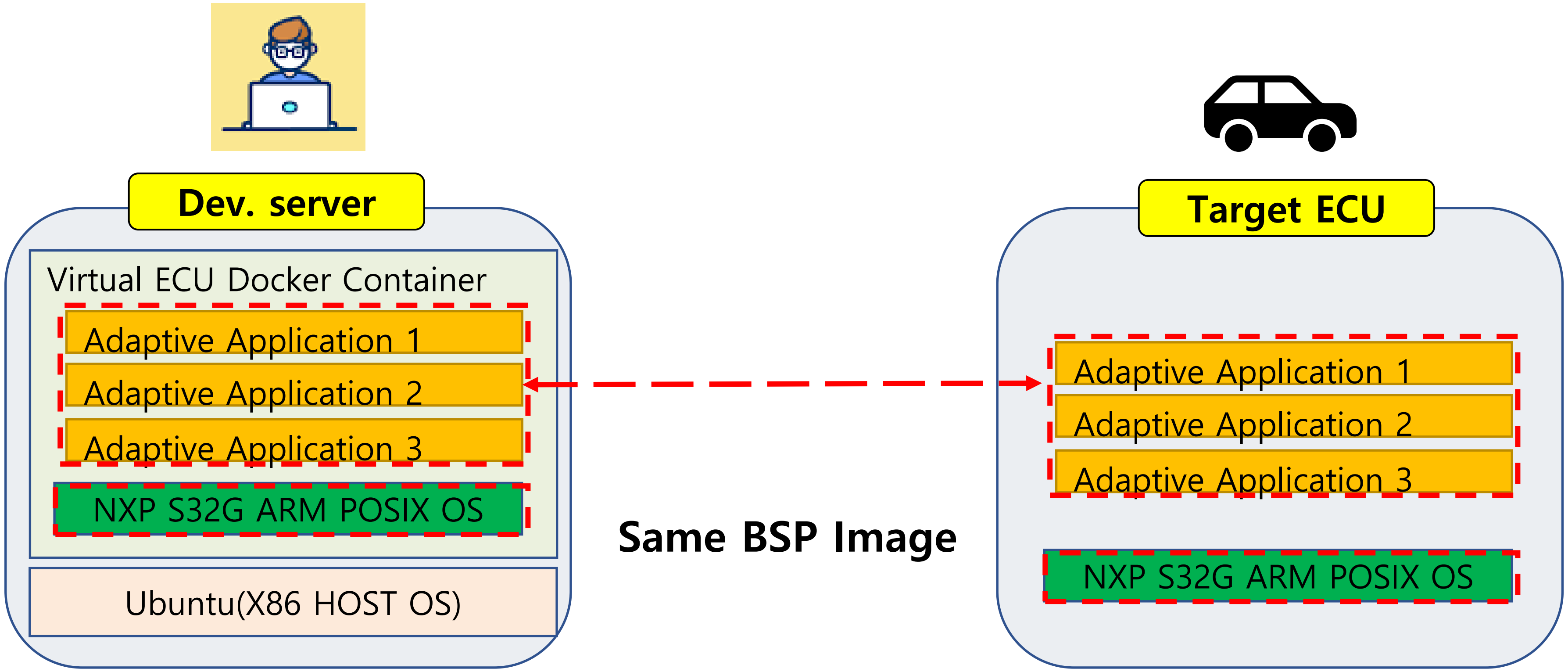
vECU_Variant3:1.0.0

- Separate virtual ECUs for different variants.
- Customer's virtual ECU can be redistributed & managed through Jenkins CI/CD.
- Virtual ECU is compatible with PopcornSAR's/other ECUs.

Jenkins

Auto-distribution
through Jenkins

5. Product introduction: Virtual ECU – Identical environment to target ECU



In case of QNX, virtual ECU will run on QEMU instead of docker container.

- Once testing is done in a development server, an adaptive application can be ported to target ECU without recompiling.

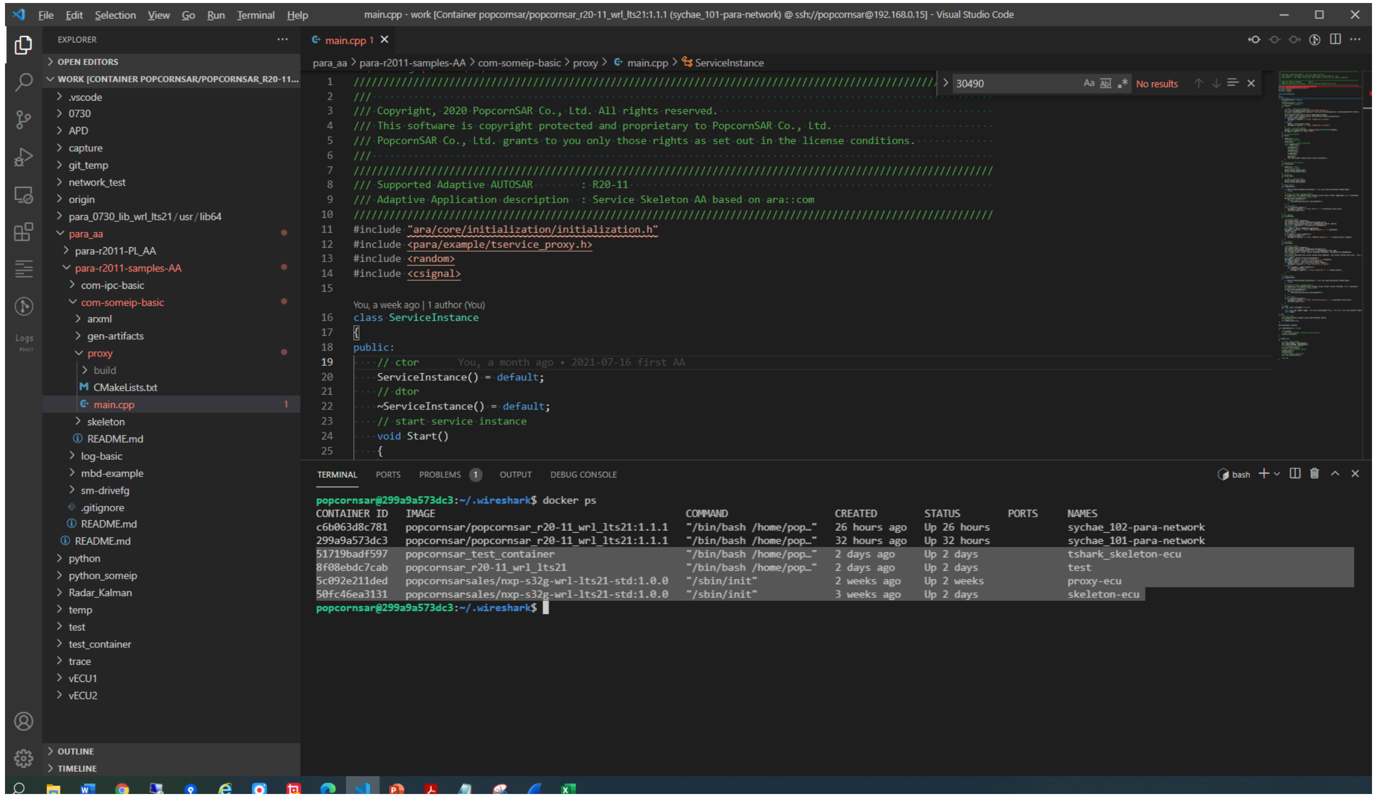
  -> Maximizes efficiency in development.

5. Product introduction: Virtual ECU - QEMU vs Docker Container

	QEMU (AS-IS)	DockerContainer(To-be)
Real-time simulation for adaptive application in target board	X (Consuming ROM/RAM)	O
Virtual ECU configuration management	X	O
Sharing with others	X	O
Distribution through network	X	O
Adding adaptive application in virtual ECU	X	O
Others	-	Can connect to other docker containers for system tests

5. Product introduction: Virtual ECU - Examples

- **Developer can test network by generating multiple virtual ECUs.**

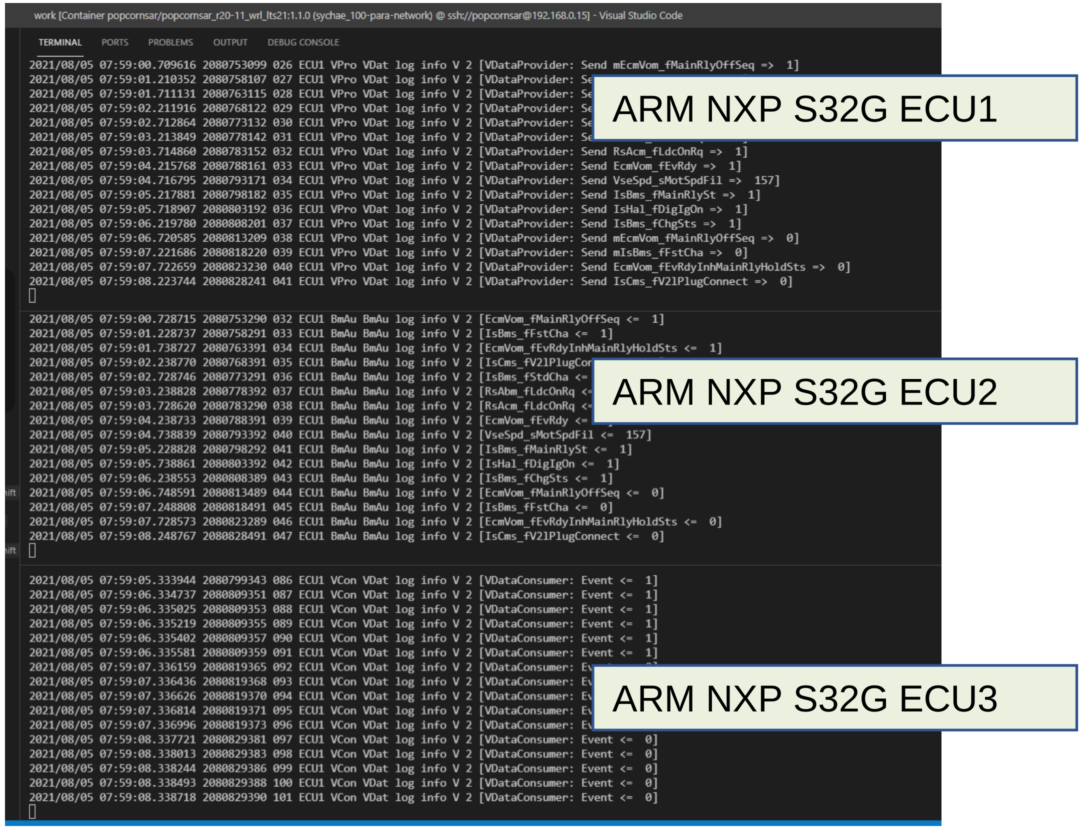


```cpp
/////////////////////////////////////////////////////////////////////////////////////
///
/// Copyright, 2020 PopcornSAR Co., Ltd. All rights reserved.
/// This software is copyright protected and proprietary to PopcornSAR Co., Ltd.
/// PopcornSAR Co., Ltd. grants to you only those rights as set out in the license conditions.
///
/////////////////////////////////////////////////////////////////////////////////////
/// Supported Adaptive AUTOSAR         : R20-11
/// Adaptive Application description   : Service Skeleton AA based on ara::com
/////////////////////////////////////////////////////////////////////////////////////
#include "ara/core/initialization/initialization.h"
#include <para/example/tservice_proxy.h>
#include <random>
#include <csignal>

class ServiceInstance
{
public:
    // ctor       You, a month ago • 2021-07-16 first AA
    ServiceInstance() = default;
    // dtor
    ~ServiceInstance() = default;
    // start service instance
    void Start()
    {
```

```
popcornsar@299a9a573dc3:~/.wireshark$ docker ps
CONTAINER ID   IMAGE                                        COMMAND                  CREATED        STATUS         PORTS     NAMES
c6b063d8c781   popcornsar/popcornsar_r20-11_wrl_lts21:1.1.1   "/bin/bash /home/pop…"   26 hours ago   Up 26 hours              sychae_102-para-network
299a9a573dc3   popcornsar/popcornsar_r20-11_wrl_lts21:1.1.1   "/bin/bash /home/pop…"   32 hours ago   Up 32 hours              sychae_101-para-network
51719badf597   popcornsar_test_container                    "/bin/bash /home/pop…"   2 days ago     Up 2 days                tshark_skeleton-ecu
8f08ebdc7cab   popcornsar_r20-11_wrl_lts21                  "/bin/bash /home/pop…"   2 days ago     Up 2 days                test
5c092e211ded   popcornsarsales/nxp-s32g-wrl-lts21-std:1.0.0   "/sbin/init"             2 weeks ago    Up 2 weeks               proxy-ecu
50fc46ea3131   popcornsarsales/nxp-s32g-wrl-lts21-std:1.0.0   "/sbin/init"             3 weeks ago    Up 2 days                skeleton-ecu
popcornsar@299a9a573dc3:~/.wireshark$
```

<Virtual ECU in PACON IDE>


work [Container popcornsar/popcornsar_r20-11_wrl_lts21:1.1.0 (sychae_100-para-network) @ ssh://popcornsar@192.168.0.15] - Visual Studio Code
TERMINAL
PORTS
PROBLEMS
OUTPUT
DEBUG CONSOLE

ARM NXP S32G ECU1

```
2021/08/05 07:59:00.709616 2080753099 026 ECU1 VPro VDat log info V 2 [VDataProvider: Send mEcmVom_fMainRlyOffSeq =>  1]
2021/08/05 07:59:01.210352 2080758107 027 ECU1 VPro VDat log info V 2 [VDataProvider: Se
2021/08/05 07:59:01.711131 2080763115 028 ECU1 VPro VDat log info V 2 [VDataProvider: Se
2021/08/05 07:59:02.211916 2080768122 029 ECU1 VPro VDat log info V 2 [VDataProvider: Se
2021/08/05 07:59:02.712864 2080773132 030 ECU1 VPro VDat log info V 2 [VDataProvider: Se
2021/08/05 07:59:03.213849 2080778142 031 ECU1 VPro VDat log info V 2 [VDataProvider: Se
2021/08/05 07:59:03.714860 2080783152 032 ECU1 VPro VDat log info V 2 [VDataProvider: Send RsAcm_fLdcOnRq =>  1]
2021/08/05 07:59:04.215768 2080788161 033 ECU1 VPro VDat log info V 2 [VDataProvider: Send EcmVom_fEvRdy =>  1]
2021/08/05 07:59:04.716795 2080793171 034 ECU1 VPro VDat log info V 2 [VDataProvider: Send VseSpd_sMotSpdFil =>  157]
2021/08/05 07:59:05.217881 2080798182 035 ECU1 VPro VDat log info V 2 [VDataProvider: Send IsBms_fMainRlySt =>  1]
2021/08/05 07:59:05.718907 2080803192 036 ECU1 VPro VDat log info V 2 [VDataProvider: Send IsHal_fDigIgOn =>  1]
2021/08/05 07:59:06.219780 2080808201 037 ECU1 VPro VDat log info V 2 [VDataProvider: Send IsBms_fChgSts =>  1]
2021/08/05 07:59:06.720585 2080813209 038 ECU1 VPro VDat log info V 2 [VDataProvider: Send mEcmVom_fMainRlyOffSeq =>  0]
2021/08/05 07:59:07.221686 2080818220 039 ECU1 VPro VDat log info V 2 [VDataProvider: Send mIsBms_fFstCha =>  0]
2021/08/05 07:59:07.722659 2080823230 040 ECU1 VPro VDat log info V 2 [VDataProvider: Send EcmVom_fEvRdyInhMainRlyHoldSts =>  0]
2021/08/05 07:59:08.223744 2080828241 041 ECU1 VPro VDat log info V 2 [VDataProvider: Send IsCms_fV21PlugConnect =>  0]
▯
```

ARM NXP S32G ECU2

```
2021/08/05 07:59:00.728715 2080753290 032 ECU1 BmAu BmAu log info V 2 [EcmVom_fMainRlyOffSeq <=  1]
2021/08/05 07:59:01.228737 2080758291 033 ECU1 BmAu BmAu log info V 2 [IsBms_fFstCha <=  1]
2021/08/05 07:59:01.738727 2080763391 034 ECU1 BmAu BmAu log info V 2 [EcmVom_fEvRdyInhMainRlyHoldSts <=  1]
2021/08/05 07:59:02.238770 2080768391 035 ECU1 BmAu BmAu log info V 2 [IsCms_fV21PlugCor
2021/08/05 07:59:02.728746 2080773291 036 ECU1 BmAu BmAu log info V 2 [IsBms_fStdCha <=
2021/08/05 07:59:03.238828 2080778392 037 ECU1 BmAu BmAu log info V 2 [RsAbm_fLdcOnRq <=
2021/08/05 07:59:03.728620 2080783290 038 ECU1 BmAu BmAu log info V 2 [RsAcm_fLdcOnRq <=
2021/08/05 07:59:04.238733 2080788391 039 ECU1 BmAu BmAu log info V 2 [EcmVom_fEvRdy <=
2021/08/05 07:59:04.738839 2080793392 040 ECU1 BmAu BmAu log info V 2 [VseSpd_sMotSpdFil <=  157]
2021/08/05 07:59:05.228828 2080798292 041 ECU1 BmAu BmAu log info V 2 [IsBms_fMainRlySt <=  1]
2021/08/05 07:59:05.738861 2080803392 042 ECU1 BmAu BmAu log info V 2 [IsHal_fDigIgOn <=  1]
2021/08/05 07:59:06.238553 2080808389 043 ECU1 BmAu BmAu log info V 2 [IsBms_fChgSts <=  1]
2021/08/05 07:59:06.748591 2080813489 044 ECU1 BmAu BmAu log info V 2 [EcmVom_fMainRlyOffSeq <=  0]
2021/08/05 07:59:07.248808 2080818491 045 ECU1 BmAu BmAu log info V 2 [IsBms_fFstCha <=  0]
2021/08/05 07:59:07.728573 2080823289 046 ECU1 BmAu BmAu log info V 2 [EcmVom_fEvRdyInhMainRlyHoldSts <=  0]
2021/08/05 07:59:08.248767 2080828491 047 ECU1 BmAu BmAu log info V 2 [IsCms_fV21PlugConnect <=  0]
▯
```

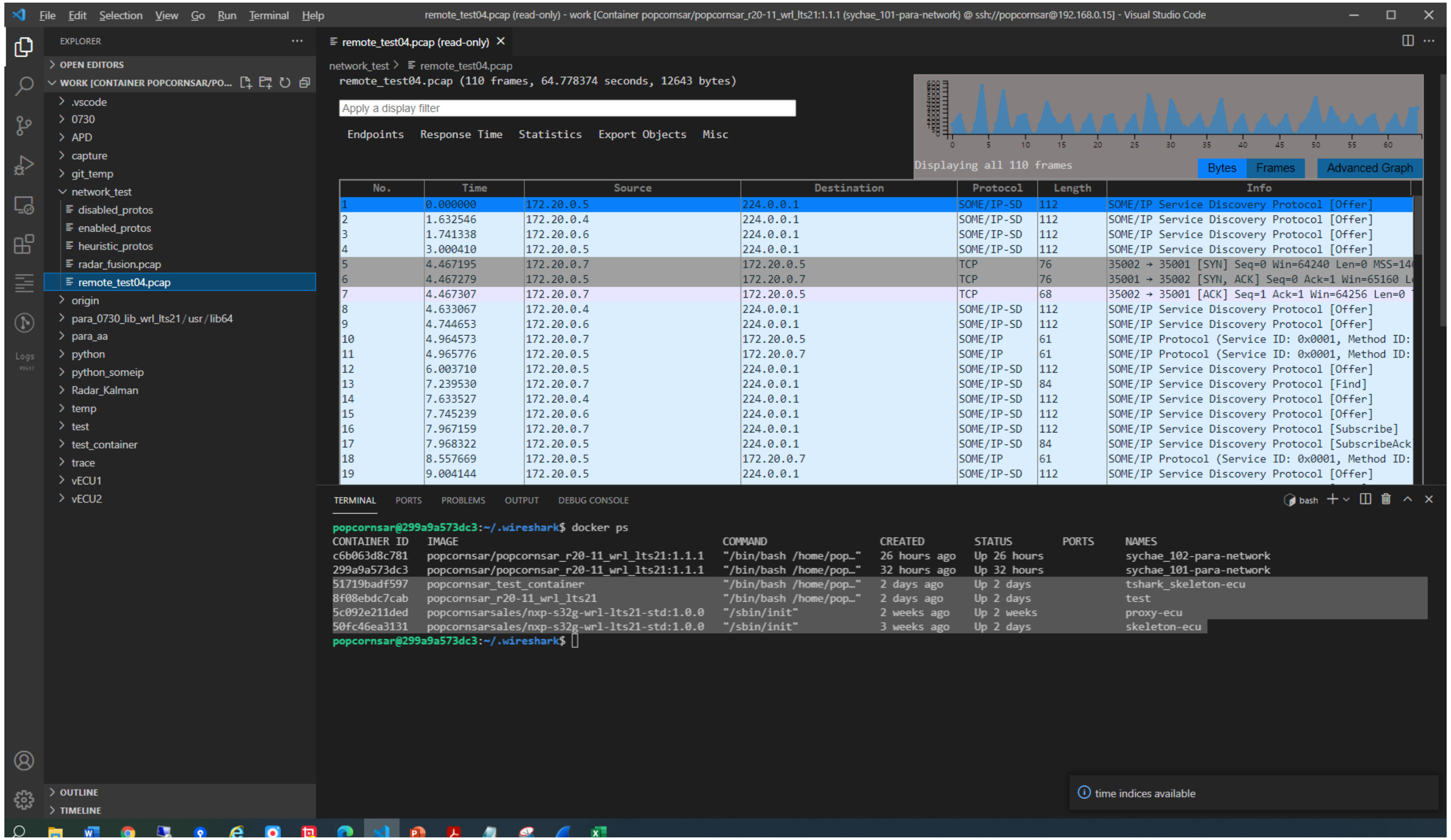
ARM NXP S32G ECU3

```
2021/08/05 07:59:05.333944 2080799343 086 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  1]
2021/08/05 07:59:06.334737 2080809351 087 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  1]
2021/08/05 07:59:06.335025 2080809353 088 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  1]
2021/08/05 07:59:06.335219 2080809355 089 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  1]
2021/08/05 07:59:06.335402 2080809357 090 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  1]
2021/08/05 07:59:06.335581 2080809359 091 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  1]
2021/08/05 07:59:07.336159 2080819365 092 ECU1 VCon VDat log info V 2 [VDataConsumer: Ev
2021/08/05 07:59:07.336436 2080819368 093 ECU1 VCon VDat log info V 2 [VDataConsumer: Ev
2021/08/05 07:59:07.336626 2080819370 094 ECU1 VCon VDat log info V 2 [VDataConsumer: Ev
2021/08/05 07:59:07.336814 2080819371 095 ECU1 VCon VDat log info V 2 [VDataConsumer: Ev
2021/08/05 07:59:07.336996 2080819373 096 ECU1 VCon VDat log info V 2 [VDataConsumer: Ev
2021/08/05 07:59:08.337721 2080829381 097 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  0]
2021/08/05 07:59:08.338013 2080829383 098 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  0]
2021/08/05 07:59:08.338244 2080829386 099 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  0]
2021/08/05 07:59:08.338493 2080829388 100 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  0]
2021/08/05 07:59:08.338718 2080829390 101 ECU1 VCon VDat log info V 2 [VDataConsumer: Event <=  0]
▯
```

<Terminal in PACON IDE>

5. Product introduction: Virtual ECU - Examples

- **Can use tshark in Docker Container for system test to monitor virtual ECU(ARM, etc.).**
- **Can use Wireshark which is installed in PACON IDE.**



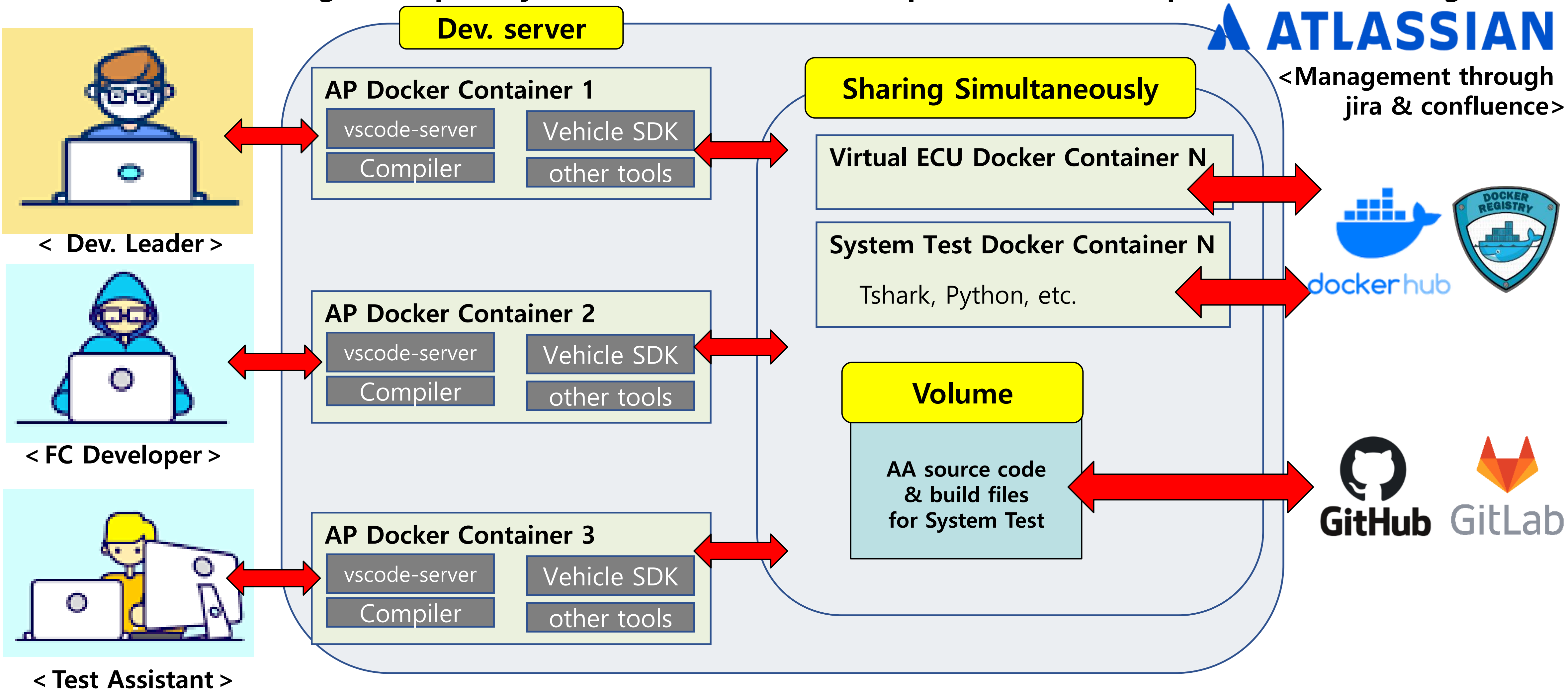
remote_test04.pcap (110 frames, 64.778374 seconds, 12643 bytes)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.20.0.5	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]
2	1.632546	172.20.0.4	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]
3	1.741338	172.20.0.6	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]
4	3.000410	172.20.0.5	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]
5	4.467195	172.20.0.7	172.20.0.5	TCP	76	35002 → 35001 [SYN] Seq=0 Win=64240 Len=0 MSS=14
6	4.467279	172.20.0.5	172.20.0.7	TCP	76	35001 → 35002 [SYN, ACK] Seq=0 Ack=1 Win=65160 L
7	4.467307	172.20.0.7	172.20.0.5	TCP	68	35002 → 35001 [ACK] Seq=1 Ack=1 Win=64256 Len=0
8	4.633067	172.20.0.4	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]
9	4.744653	172.20.0.6	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]
10	4.964573	172.20.0.7	172.20.0.5	SOME/IP	61	SOME/IP Protocol (Service ID: 0x0001, Method ID:
11	4.965776	172.20.0.5	172.20.0.7	SOME/IP	61	SOME/IP Protocol (Service ID: 0x0001, Method ID:
12	6.003710	172.20.0.5	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]
13	7.239530	172.20.0.7	224.0.0.1	SOME/IP-SD	84	SOME/IP Service Discovery Protocol [Find]
14	7.633527	172.20.0.4	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]
15	7.745239	172.20.0.6	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]
16	7.967159	172.20.0.7	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Subscribe]
17	7.968322	172.20.0.5	224.0.0.1	SOME/IP-SD	84	SOME/IP Service Discovery Protocol [SubscribeAck
18	8.557669	172.20.0.5	172.20.0.7	SOME/IP	61	SOME/IP Protocol (Service ID: 0x0001, Method ID:
19	9.004144	172.20.0.5	224.0.0.1	SOME/IP-SD	112	SOME/IP Service Discovery Protocol [Offer]

```
popcornsar@299a9a573dc3:~/.wireshark$ docker ps
CONTAINER ID   IMAGE                                     COMMAND                  CREATED         STATUS          PORTS     NAMES
c6b063d8c781   popcornsar/popcornsar_r20-11_wrl_lts21:1.1.1   "/bin/bash /home/pop…"   26 hours ago    Up 26 hours               sychae_102-para-network
299a9a573dc3   popcornsar/popcornsar_r20-11_wrl_lts21:1.1.1   "/bin/bash /home/pop…"   32 hours ago    Up 32 hours               sychae_101-para-network
51719badf597   popcornsar_test_container                 "/bin/bash /home/pop…"   2 days ago      Up 2 days                 tshark_skeleton-ecu
8f08ebdc7cab   popcornsar_r20-11_wrl_lts21               "/bin/bash /home/pop…"   2 days ago      Up 2 days                 test
5c092e211ded   popcornsarsales/nxp-s32g-wrl-lts21-std:1.0.0   "/sbin/init"             2 weeks ago     Up 2 weeks                proxy-ecu
50fc46ea3131   popcornsarsales/nxp-s32g-wrl-lts21-std:1.0.0   "/sbin/init"             3 weeks ago     Up 2 days                 skeleton-ecu
popcornsar@299a9a573dc3:~/.wireshark$ 
```

6. Example of PopcornSAR AP development environment

- PopcornSAR AP tool chain doesn’t include QEMU nor Yocto.
- Since ARA API changes frequently for each AP version, PopcornSAR developed its own testing SW.



< Dev. Leader >

< FC Developer >

< Test Assistant >

Dev. server

AP Docker Container 1

vscode-server

Compiler

Vehicle SDK

other tools

AP Docker Container 2

vscode-server

Compiler

Vehicle SDK

other tools

AP Docker Container 3

vscode-server

Compiler

Vehicle SDK

other tools

Sharing Simultaneously

Virtual ECU Docker Container N

System Test Docker Container N

Tshark, Python, etc.

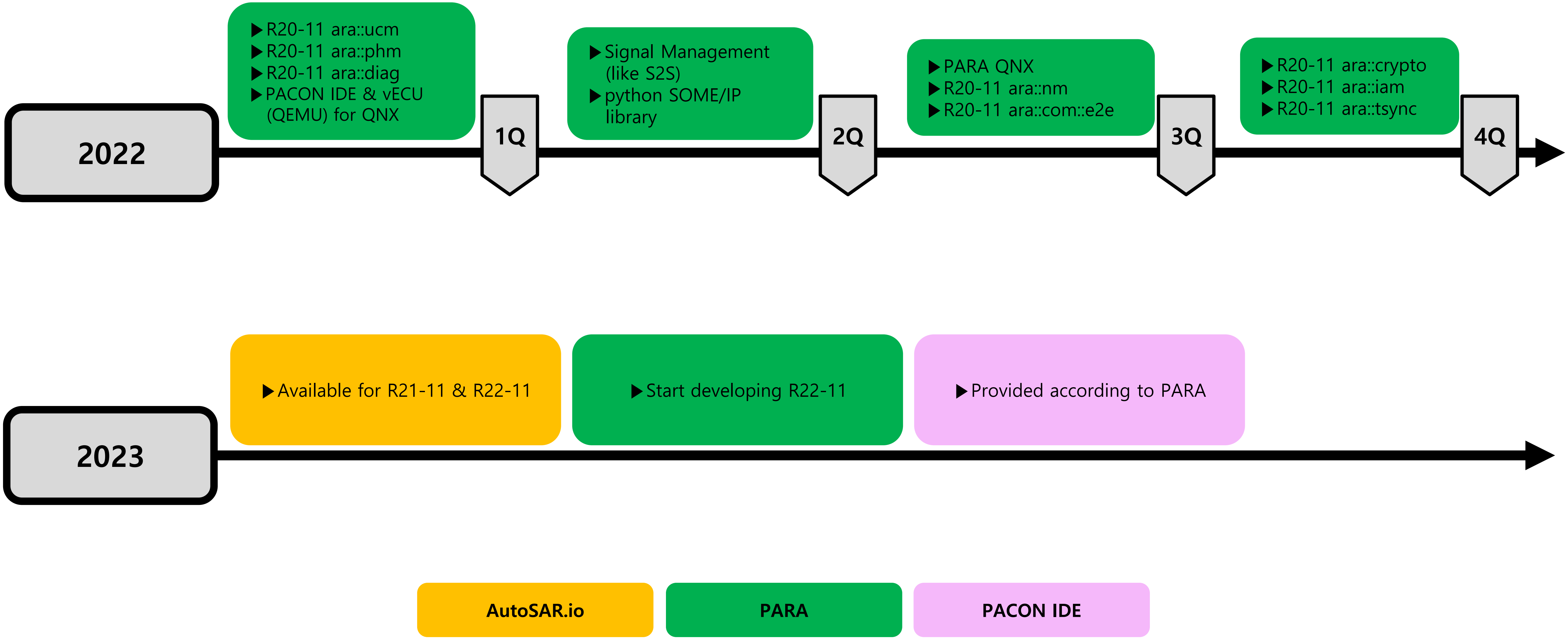
Volume

AA source code
& build files
for System Test

ATLASSIAN

<Management through
jira & confluence>

7. PopcornSAR Adaptive tool development roadmap (~2023)



2022

- ▶ R20-11 ara::ucm
- ▶ R20-11 ara::phm
- ▶ R20-11 ara::diag
- ▶ PACON IDE & vECU (QEMU) for QNX

1Q

- ▶ Signal Management (like S2S)
- ▶ python SOME/IP library

2Q

- ▶ PARA QNX
- ▶ R20-11 ara::nm
- ▶ R20-11 ara::com::e2e

3Q

- ▶ R20-11 ara::crypto
- ▶ R20-11 ara::iam
- ▶ R20-11 ara::tsync

4Q

2023

- ▶ Available for R21-11 & R22-11
- ▶ Start developing R22-11
- ▶ Provided according to PARA

AutoSAR.io

PARA

PACON IDE

(Appendix 1) Tool demonstrations

1. **R20-11 Autosar.io ARXML modeling :**

   - Link : https://youtu.be/3FYzR0bQ44s

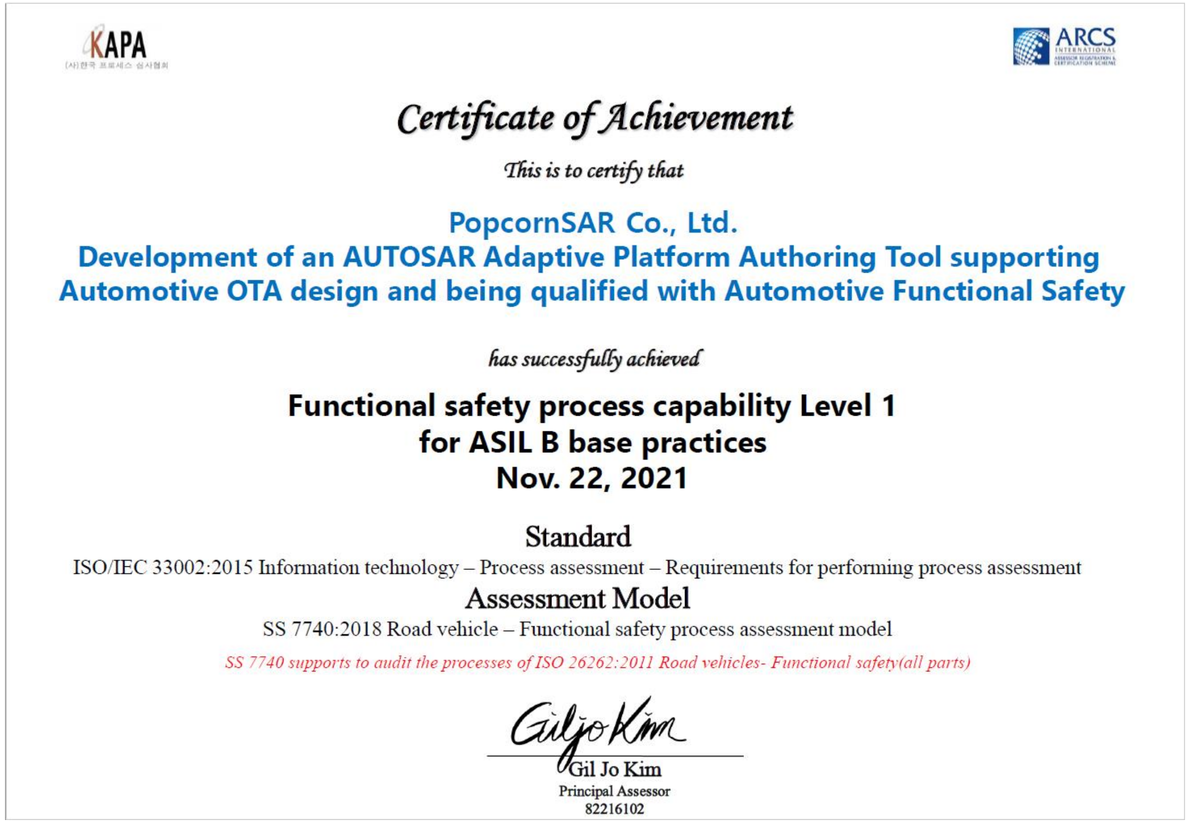
2. **R20-11 generation & build for communication between 2 Adaptive Applications :**

   - Link : https://youtu.be/HYqNEMrYYAw

3. **R20-11 ARA::DIAG demo:**

   - Link : https://youtu.be/jvySoUdoAJs
   - Link : https://youtu.be/tm_Cr80d52w

(Appendix 2) ISO26262 Certificate of PopcornSAR Adaptive platform Authoring tool



KAPA

ARCS

Certificate of Achievement

This is to certify that

PopcornSAR Co., Ltd.

Development of an AUTOSAR Adaptive Platform Authoring Tool supporting Automotive OTA design and being qualified with Automotive Functional Safety

has successfully achieved

Functional safety process capability Level 1
for ASIL B base practices
Nov. 22, 2021

Standard

ISO/IEC 33002:2015 Information technology – Process assessment – Requirements for performing process assessment

Assessment Model

SS 7740:2018 Road vehicle – Functional safety process assessment model

SS 7740 supports to audit the processes of ISO 26262:2011 Road vehicles- Functional safety(all parts)

Giljo Kim

Gil Jo Kim
Principal Assessor
82216102

Details of Achievement

Process ID	Process Name	Capability Level 1
		ASIL B base practices
ENG.4.SE	Software requirements analysis	●
ENG.5.SE	Software design	●
ENG.6.SE	Software construction	●
ENG.7.SE	Software integration test	●
ENG.8.SE	Software testing	●