

# Adaptive AUTOSAR (R20-11)

## 開発プラットフォームの紹介

# 目次

---

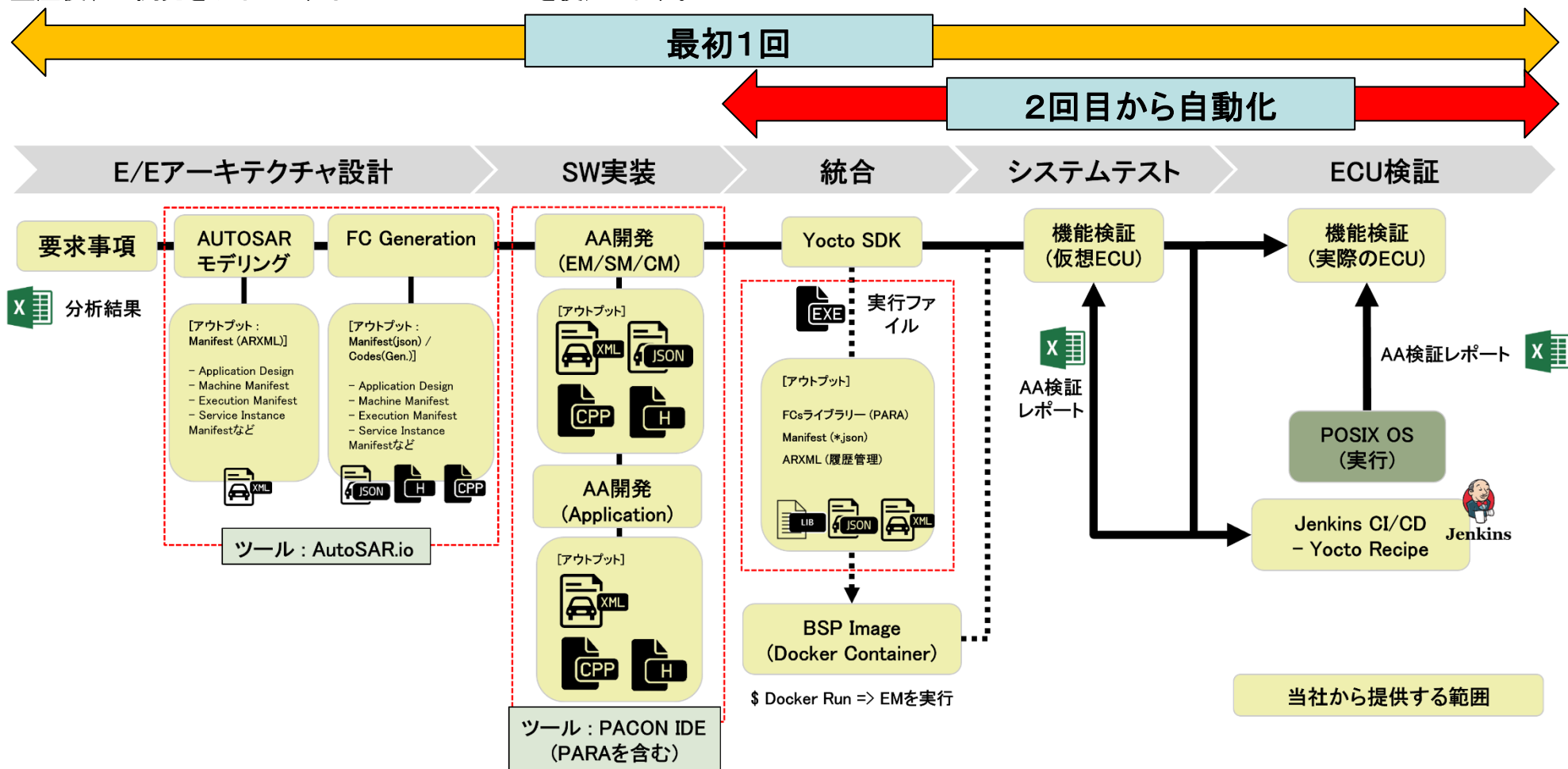
1. Adaptive AUTOSAR (R20-11)の開発フロー
2. ポップコーンザーのAdaptive AUTOSAR (R20-11)の開発プラットフォーム概要
3. 製品紹介 : PARA
4. 製品紹介 : PACON IDE
5. 製品紹介 : 仮想ECU
6. ポップコーンザーのAP開発環境の構築の例
7. AP開発ロードマップ(～2023年まで)

(補足資料1) 製品デモの動画

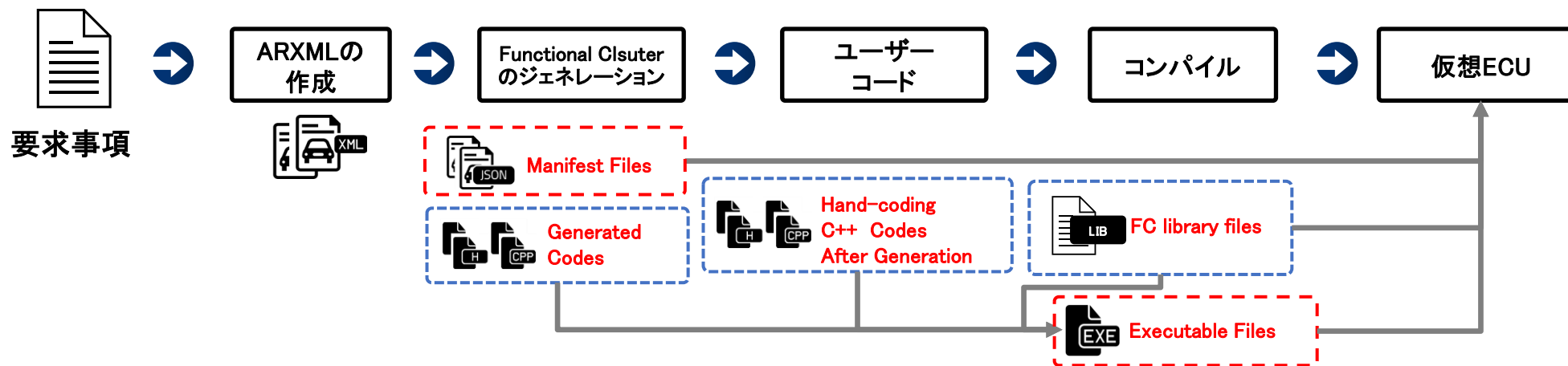
(補足資料2) 設計ツール(AutoSAR.io)のISO26262認定証

# 1. Adaptive AUTOSAR (R20-11)の開発フロー

1. 当社はEnd-to-EndでAdaptive AUTOSARツールチェーンを提供します。
  - ・ 自動車メーカーや部品メーカーのAdaptive Application/ECU開発プロセスの全てをサポートします。
2. 最初は開発フロー通りに手作業してImageを生成し、CI/CDで自動システムを構築します。
3. ECU量産後、AA開発をサポートするためにYocto SDKを使用します。



# 1. Adaptive AUTOSAR (R20-11)の開発フロー



- 仮想ECUでテストを行ってから、実際のECUでシステムテストを行います。



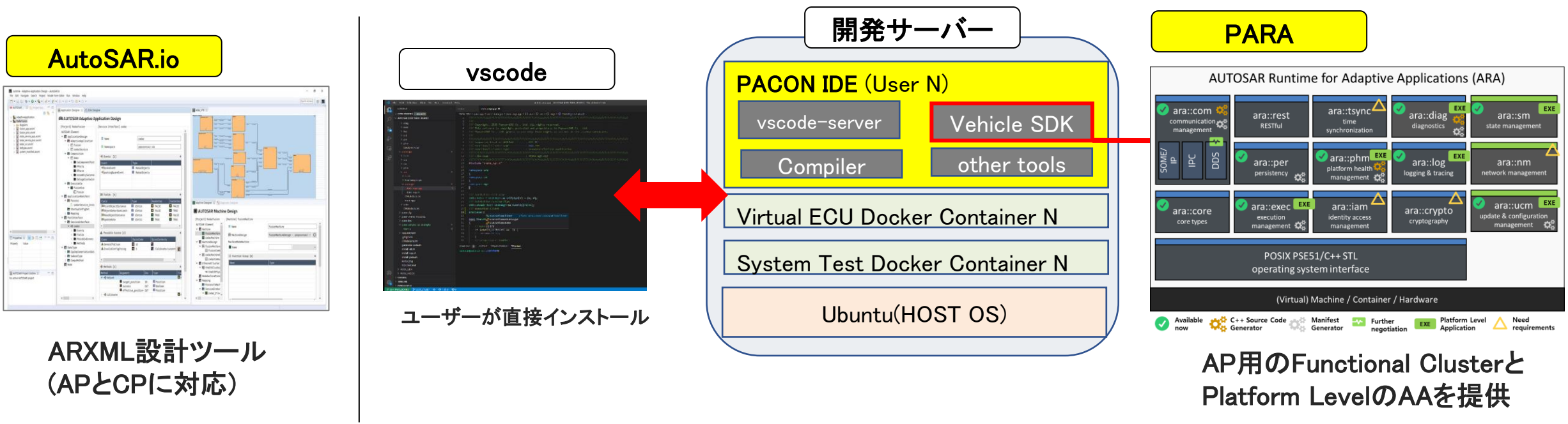
AP version (R20-11)

OTAでAPバージョンをアップグレード(R21-11)

- 量産後、機能改善やサイバーセキュリティに対応してOTAでアップデート実施

## 2. ポップコーンザーのAdaptive AUTOSAR (R20-11)の開発プラットフォーム概要

- パソコンでのインストールと連携する開発プラットフォーム
  - ① APのFC SDKまたは、Vehicle OSのSDK(ツールベンダーの依存関係なし)
  - ② 仮想ECUを提供(シミュレーションとテスト)、C++言語以外の開発言語に対応
  - ③ コーディングルールのチェック、API自動補完など

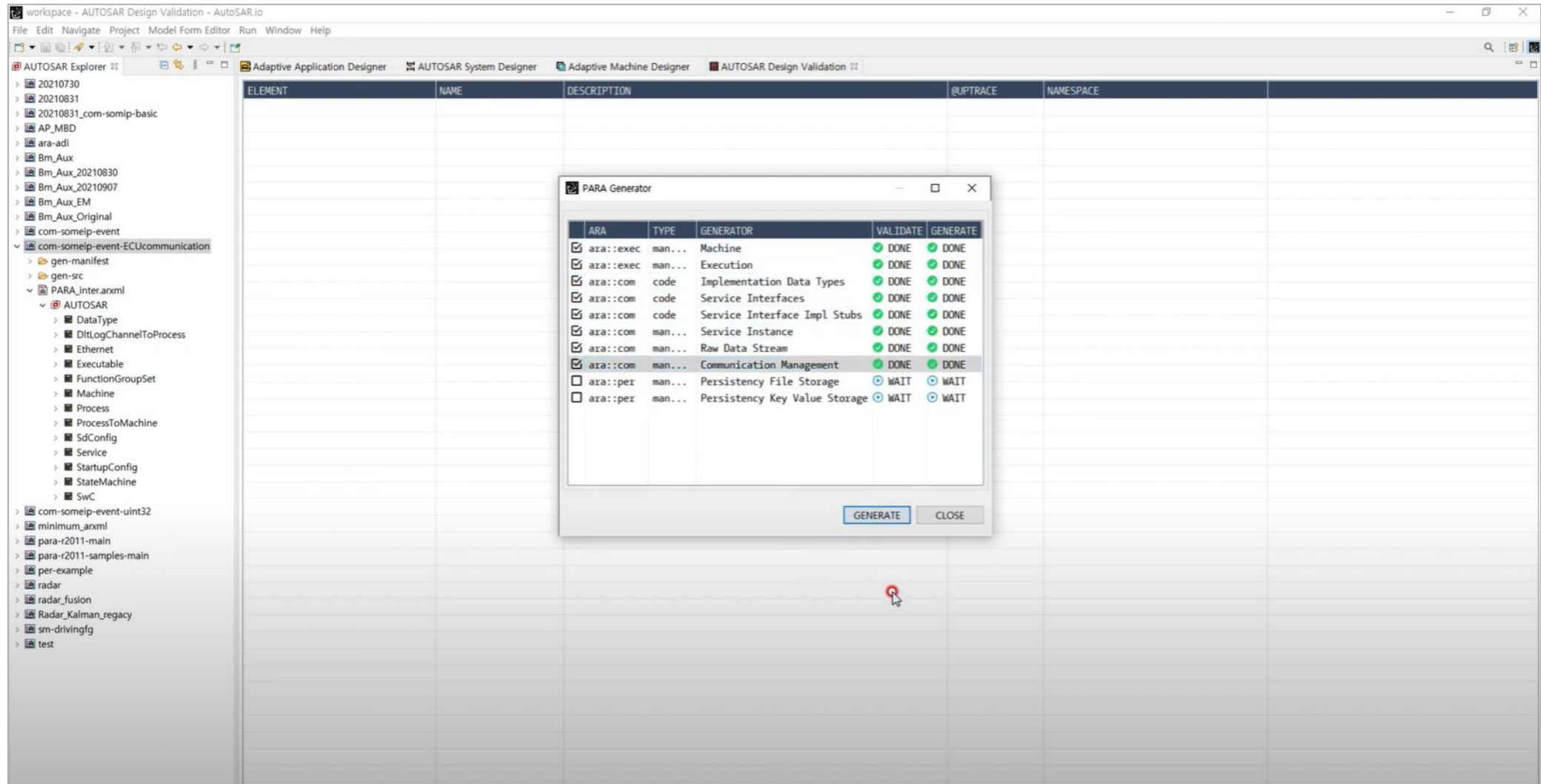


### 3. 製品紹介 : PARA - 特徴

---

1. PARAはEMなしで、AA(Adaptive Application)のテストが可能です。
  - AUTOSAR標準では、EMが必須であります。しかし、PARAでは開発の効率を高めるためにEMがなくてもAA開発やテストが行えます。
2. ポップコーンザーのFCのジェネレータはパラメーターの入力が不要です。  
ARXMLのValidationがOKになった場合、自動的にジェネレータを起動します。
  - AutoSAR.ioにはFCのジェネレータが含まれているので、自動的にAAとMachineを解釈してAAに関するソースコードとManifestファイルを生成します。
3. MATLAB/Simulink 2021a(R19-11)とAP MBD, Legacy MBDのアウトプット(C++)をAAへマイグレーションする検証を終えて、ガイドラインを提供します。
  - ユーザーが制御用AAを開発する際、必ずMATLAB/Simulinkを使用します。
4. PARAはAI開発向けのPython用SOME/IP通信に対応します。

### 3. 製品紹介 : PARA - 特徴



<AutoSAR.ioでは1分以内にPARAをジェネレーションします。>

### 3. 製品紹介 : PARA – FCリスト (R20-11)の開発計画(2023年まで)

#### AUTOSAR Runtime for Adaptive Applications (ARA)



POSIX PSE51/C++ STL  
operating system interface

(Virtual) Machine / Container / Hardware

- Available now
- Available before 2023
- C++ Source Code Generator
- Manifest Generator
- Specific Generator
- Platform Level Application
- PopcornSAR Specific



### 3. 製品紹介 : PARA – ISO26262向けの独自のSafety Mechanism

- ポップコーンザーの自社開発した機能で(libpara)、Adaptive Applicationを開発する際、ユーザーのミスでara APIを間違えて作成したソースコードやPOSIX OSで起きるHuman errorを検出する機能です。  
(232個以上の機能)

```
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,path:/example/var/ara-channel/com/4600_BMS_AA_RootSwc_RequiredPort.service <= /example/var/ara-channel/com/4600_BMS_AA_RootSwc_RequiredPort.tmp]
2021/11/19 08:08:04.216950 3612010537 102 ECU1 unde DFLT log error V 1 [CmRouter::ProcessOfferServiceReq:: not exist service fail <key:ppport: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/ara-channel/com/4600_BMS_AA_RootSwc_RequiredPort.tmp]
```

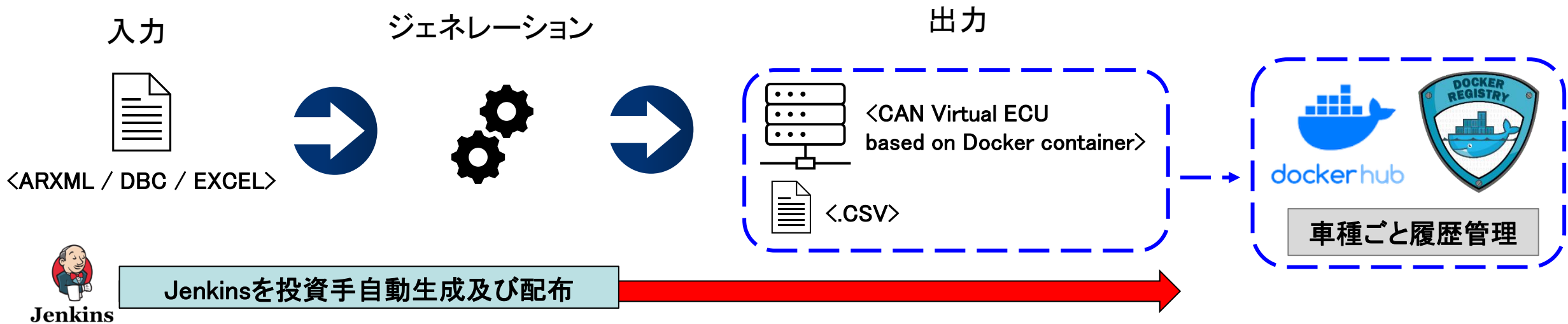
<例1. ara APIのHuman error、AA実行時、Someipエラー検出 : someip OfferService API>

```
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 #62)
```

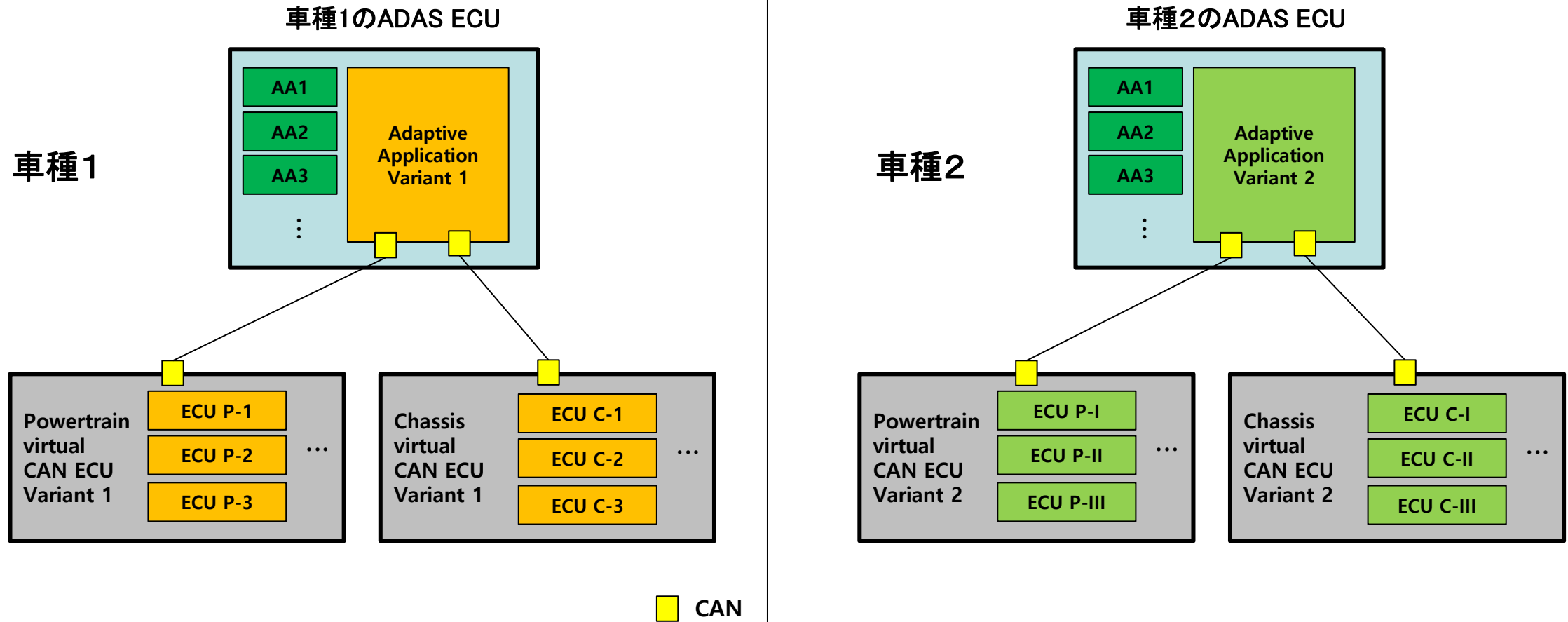
<例2. POSIX OSのHuman error、AA実行時、ボードからPOSIX OSのvi editor使用に関するdummyファイル生成の 이슈>

### 3. 製品紹介 : PARA – CAN-DO(based on Docker container)

- CAN-DOは自社開発のジェネレータです。ARXML、DBC、EXCELを入力すると、自動的にdocker containerベースのCANの仮想ECUを生成します。
- 短時間で生成された複数のCANの仮想ECUを使ってテストが行えます。
- DockerhubからCANの仮想ECUの履歴管理ができ、JenkinsからCANの仮想ECU生成を自動化します。
- 非AUTOSARにも対応します。

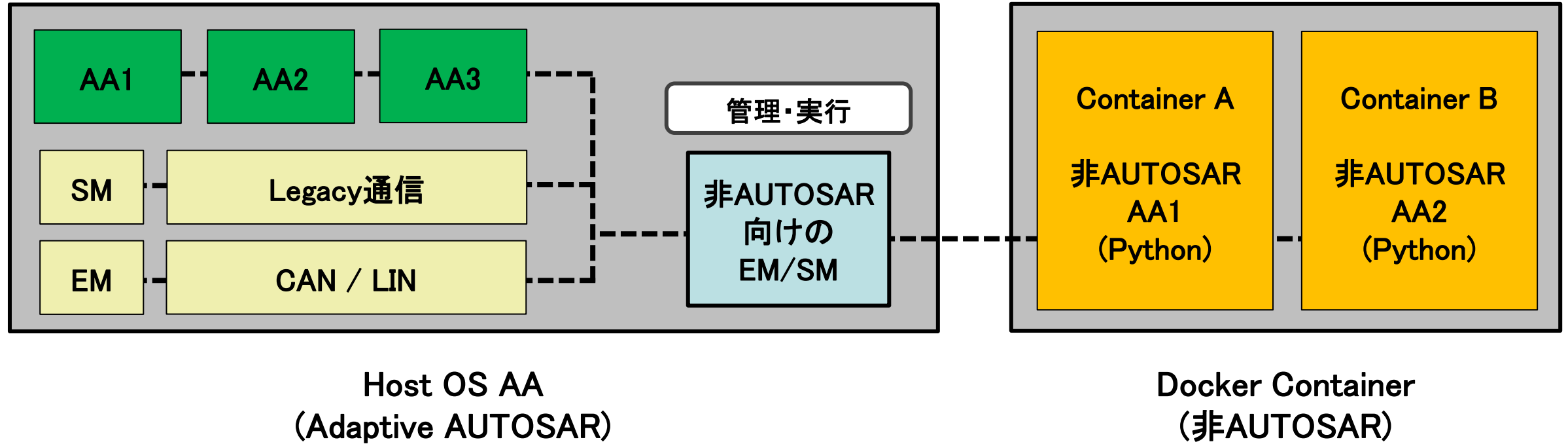


### 3. 製品紹介 : PARA – CAN-DOの使用例



- CAN-DOはVirtual CAN ECUを自動的に生成します。
- 各ECUとその設定に関する情報はDockerhubまたはDocker registryに保存・管理されます。

### 3. 製品紹介 : PARA – 非AUTOSAR Application実行のための特徴AA



- EM適用時、非AUTOSARアプリケーション(docker・Legacy通信・Pythonなど)を認識できない場合があります。
- 解決方法としてEM・SMの役割を果たせるAAを生成して、非AUTOSARアプリケーションを実行することができます。

## 4. 製品紹介 : PACON IDE – 特徴

---

1. PACON IDEをカスタマーブランドのIDEとしてカスタマイズできます。
  - 車種毎にIDEを管理することができます。
  - Jenkins CI/CDで、カスタマーブランドIDEの配布や履歴管理ができます。
2. PACON IDEはAdaptive Platform Vendorに依存関係がありません。
3. VSCodeと連携して、PACON IDE(=Docker Container)を使えます。
  - Docker ContainerにはPARAとVSCodeで使用する拡張機能が含まれています。
  - 外部ネットワークに繋がらなくても使えます。
4. PACON IDEにインストールされているWiresharkを使えます。(ユーザーのPCにインストール不要)
5. Jenkins CI/CDからPACON IDEを配布することができます。
6. 開発効率を高める機能を提供します。
  - ARA APIとオープンソースAPIの自動補完
  - リアルタイムでコーディングルールをチェック
  - デバッグ、多様なプログラミング言語(Pythonなど)に対応

## 4. 製品紹介 : PACON IDE - カスタマーブランドIDEへのカスタマイズ

| 車種        | 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       |

- 車種毎にIDEを管理することができます。
- Jenkins CI/CDで、カスタマーブランドIDEの配布や履歴管理ができます。
- OSアップデートにより、IDEを再度配布する必要があります。



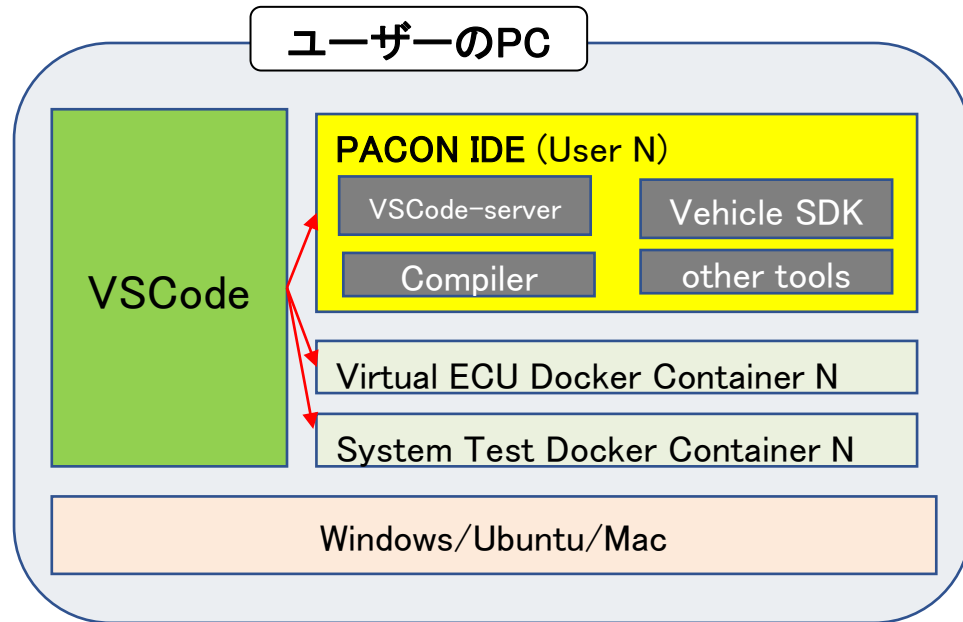
**Jenkins**

Jenkinsを通して  
自動的に配布

## 4. 製品紹介 : PACON IDE – 提供形態

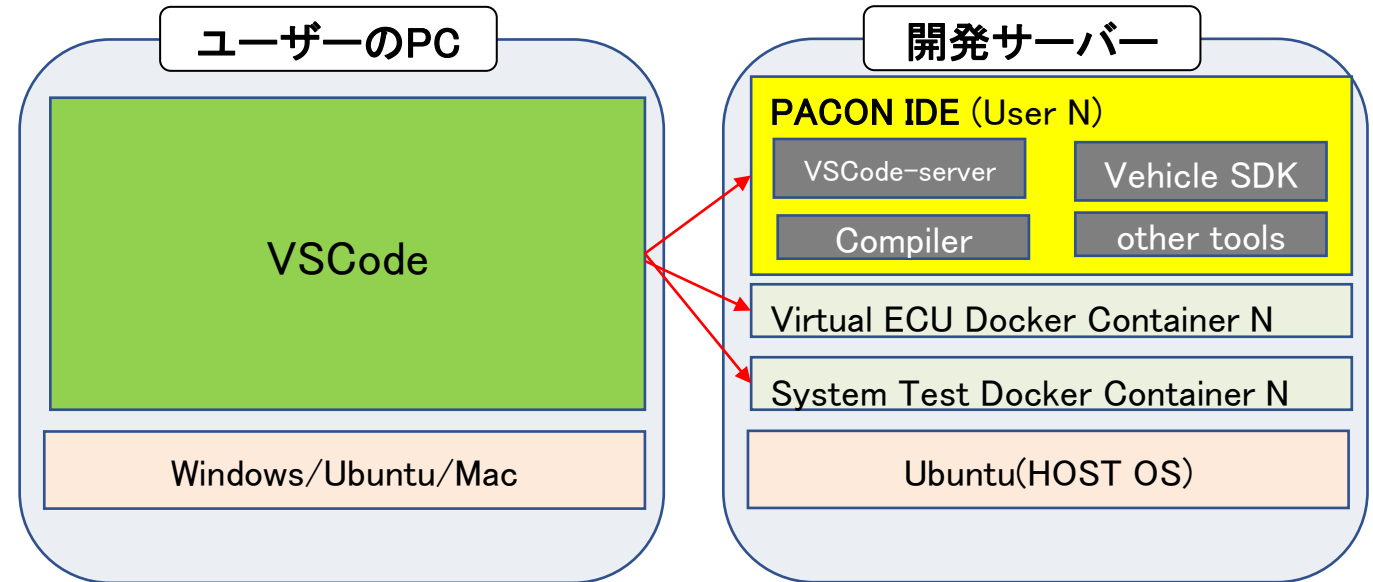
### ① ユーザーのPCから開発

(ネットワーク接続不要)



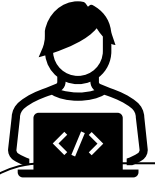
### ② ユーザーのPCが開発サーバーに連携して開発

(ユーザー同士の協業で開発を効率化させる)



docker container : dockerhubから配布する

## 4. 製品紹介 : PACON IDE – IDE及び仮想ECUの構成



ユーザーのTarget ECUの  
情報を取得

CPU  
(NXP, RENESAS, NVIDIA  
など)

使用するPOSIX OS  
(Linux, QNXなど)

使用するコンパイラ

使用するAPのFC



**Jenkins**

Target ECU用AP開発環境  
(=PACON IDE)

PACON IDE (User N)

vscode-server

Vehicle SDK

Compiler

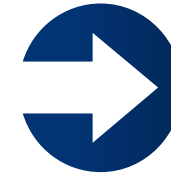
other tools

Target ECUの仮想ECU

Virtual ECU Docker Container N

システムテスト用Container

System Test Docker Container N



**Jenkins**



履歴管理



## 5. 製品紹介：仮想ECU – 特徴

---

1. カスタマーブランドの仮想ECUとしてカスタマイズできます。
  - 車種毎に仮想ECUを管理することができます。
  - Jenkins CI/CDで、カスタマーブランド仮想ECUの配布や履歴管理ができます。
2. 仮想ECUはAdaptive Platform Vendorに依存関係がありません。
3. 他社はQEMUで対応しています。(性能が低くて、互換性がありません。)
  - 当社はDocker Containerベースでカスタマーが使用するPOSIX OSとTargetECUを提供します。
  - Docker ContainerからAAのでテストが終わったら、そのままAAをTargetECUにして納品します。
4. 複数のECUをDocker Containerで開発し、DockerHubまたはDocker registryから履歴管理ができるので、ユーザー同士の共同作業に優れています。
5. Jenkins CI/CDから複数の仮想ECUを自動生成してAAテストが行えます。
6. VSCodeと連携するPACON IDEを通して、仮想ECUを生成することができます。
7. 仮想ECUと連携する他のシステムテスト用のDocker Containerを提供します。
  - 仮想ECU内にはシステムシステムSWを内蔵することが厳しい  
例：仮想ECU内に(ARM) tsharkを内蔵
  - テストSW(pythonなど)は他のシステムテスト用のDocker Container内に入れる

## 5. 製品紹介：仮想ECU – カスタマーブランドの仮想ECUへカスタマイズ

| 車種        | CPU    | OS           |                             |
|-----------|--------|--------------|-----------------------------|
| Variant 1 | NVIDIA | Linux (5.0)  | → vECU_Variant1:1.0.0       |
|           | NVIDIA | Linux (5.15) | → vECU_Variant1:1.1.0       |
| Variant 2 | Intel  | QNX          | → vECU_Variant2_intel:1.0.0 |
|           | NXP    | QNX          | → vECU_Variant2_nxp:1.0.0   |
| Variant 3 | R-car  | Android      | → vECU_Variant3:1.0.0       |

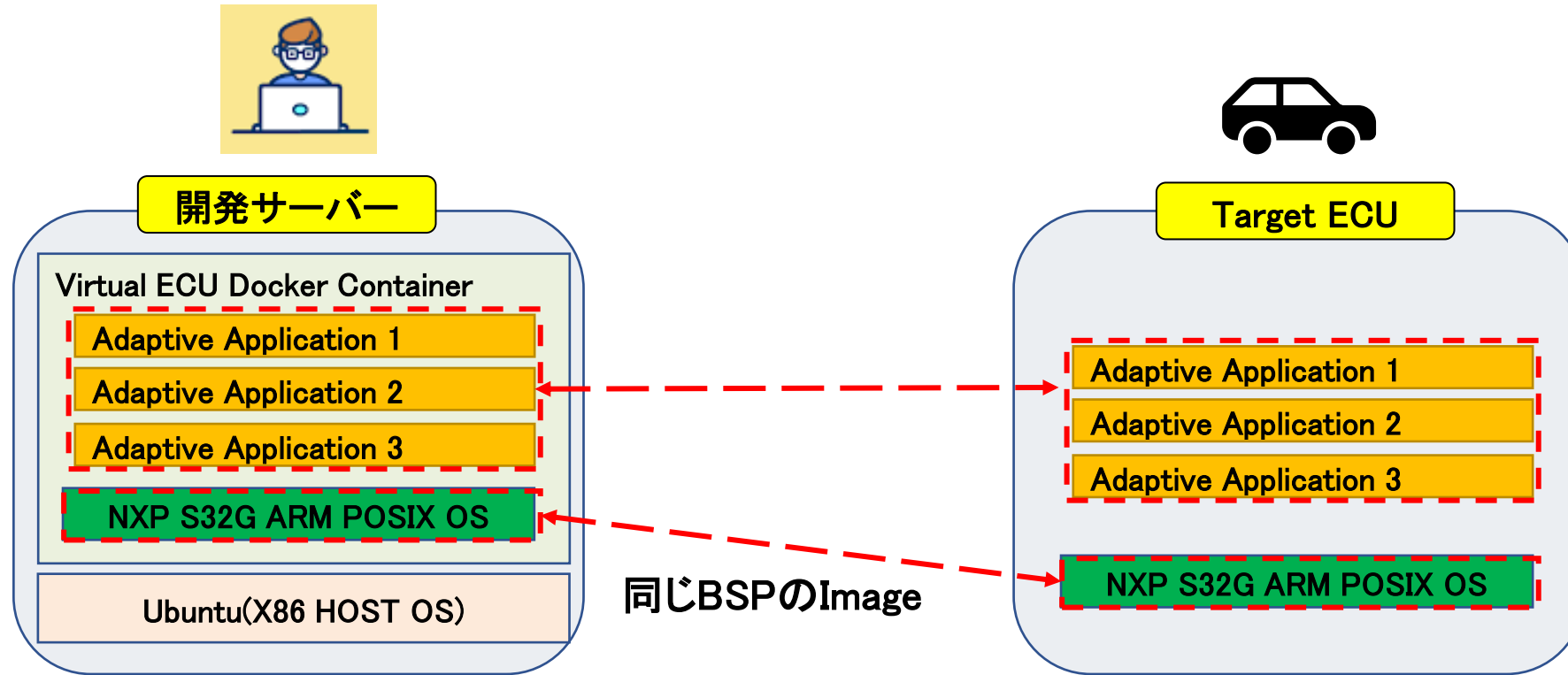
- 車種毎に仮想ECUを管理することができます。
- Jenkins CI/CDで、カスタマーブランド仮想ECUの配布や履歴管理ができます。
- 仮想ECUは自社ECUだけではなく、他社のECUとの連携まで考える。



**Jenkins**

Jenkinsを通して  
自動的に配布

## 5. 製品紹介：仮想ECU – カスタマーブランドの仮想ECUへカスタマイズ



ただ, QNXの場合はQEMUで仮想ECUを実行できる。docker containerは使いません。

- Docker ContainerでAAのテストが終わったら、再度コンパイルしなくてそのままAAをTarget ECUにして納品します。
- 再度コンパイルする必要がないので、開発にかかる時間が短縮できます。

## 5. 製品紹介：仮想ECU – QEMU 対 Docker Container

|                               | QEMU                | Docker Container                 |
|-------------------------------|---------------------|----------------------------------|
| Target boardのAA<br>リアルタイム性の保証 | X<br>(ROM/RAM使用量が高) | O                                |
| 仮想ECUの履歴管理                    | X                   | O                                |
| ユーザー同士で共有                     | X                   | O                                |
| ネットワークを介して配布                  | X                   | O                                |
| 仮想ECUにAAを<br>追加した後で配布         | X                   | O                                |
| その他                           | -                   | システムテスト用の<br>docker containerと連携 |

# 5. 製品紹介：仮想ECU – 使用例

- 複数のECUを生成して、仮想ECU間のネットワークテストが行えます。

```
main.cpp | x
1 /////////////////////////////////////////////////////////////////////////////////////////////////////////////////// | 30490
2 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
3 /// Copyright, 2020 PopcornSAR Co., Ltd. All rights reserved.
4 /// This software is copyright protected and proprietary to PopcornSAR Co., Ltd.
5 /// PopcornSAR Co., Ltd. grants to you only those rights as set out in the license conditions.
6 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
7 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
8 /// Supported Adaptive AUTOSAR : R20-11
9 /// Adaptive Application description : Service Skeleton AA based on ara::com
10 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////
11 #include "ara/core/initialization/initialization.h"
12 #include "para/example/tsservice_proxy.h"
13 #include <random>
14 #include <csignal>
15
16 You a week ago | author (You)
17 class ServiceInstance
18 {
19 public:
20 // ctor You a month ago * 2021-07-16 First AA
21 ServiceInstance() = default;
22 // dtor
23 ~ServiceInstance() = default;
24 // start service instance
25 void Start()
26 {
27
28 }
29
30 }
```

```
popcornsar@299a9a573dc3:~/wireshark$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS          NAMES
c6b06348c781  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    test            ts shark_skeleton-ecu
8f88ebdc7cab  popcornsar_r20-11_wrl_lts21            "/bin/bash /home/pop..."  2 days ago    Up 2 days    test            test
5c092e211ded  popcornsar/sales/nxp-s32g-wrl-lts21-std:1.0.0  "/sbin/init"            2 weeks ago   Up 2 weeks   proxy-ecu       proxy-ecu
58fc46ea3131  popcornsar/sales/nxp-s32g-wrl-lts21-std:1.0.0  "/sbin/init"            3 weeks ago   Up 2 days    skeleton-ecu    skeleton-ecu
```

＜PAICON IDE内の仮想ECU＞

```
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
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 07:59:03.213840 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: Se
2021/08/05 07:59:04.215768 2080788161 033 ECU1 VPro Vdat log info V 2 [VdataProvider: Se
2021/08/05 07:59:04.716795 2080793171 034 ECU1 VPro Vdat log info V 2 [VdataProvider: Se
2021/08/05 07:59:05.217881 2080798182 035 ECU1 VPro Vdat log info V 2 [VdataProvider: Se
2021/08/05 07:59:05.718907 2080803192 036 ECU1 VPro Vdat log info V 2 [VdataProvider: Se
2021/08/05 07:59:06.219780 2080808201 037 ECU1 VPro Vdat log info V 2 [VdataProvider: Se
2021/08/05 07:59:06.720585 2080813209 038 ECU1 VPro Vdat log info V 2 [VdataProvider: Se
2021/08/05 07:59:07.221686 2080818220 039 ECU1 VPro Vdat log info V 2 [VdataProvider: Se
2021/08/05 07:59:07.722659 2080823230 040 ECU1 VPro Vdat log info V 2 [VdataProvider: Se
2021/08/05 07:59:08.223744 2080828241 041 ECU1 VPro Vdat log info V 2 [VdataProvider: Se
[]
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 07:59:02.738745 2080773391 036 ECU1 BmAu BmAu log info V 2 [IsBms_ffstCha <=
2021/08/05 07:59:03.238828 2080778392 037 ECU1 BmAu BmAu log info V 2 [RsAcM_fldcOnRq <=
2021/08/05 07:59:03.738820 2080783390 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 07:59:05.238828 2080798292 041 ECU1 BmAu BmAu log info V 2 [IsBms_ffstCha <= 1]
2021/08/05 07:59:05.738861 2080803392 042 ECU1 BmAu BmAu log info V 2 [IsHa1_fDigIqOn <= 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 07:59:07.748573 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]
[]
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]
[]
```

＜PAICON IDE内のTerminal＞

## 5. 製品紹介：仮想ECU – 使用例

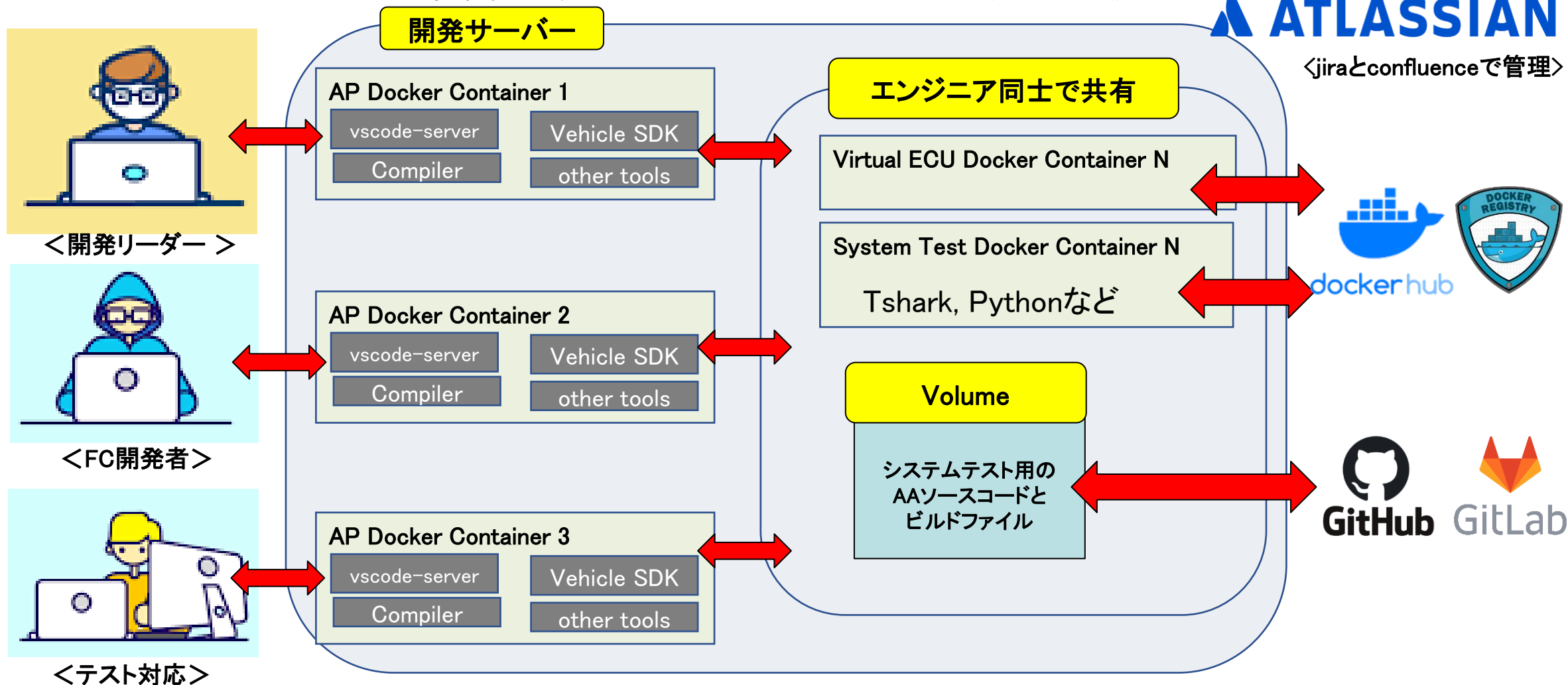
- システムテスト用のDocker Containerからtsharkを用いて仮想ECU(ARMなど)のモニタリングできます。
- PACON IDEにインストールされているWiresharkを使えます。(ユーザーのPCにインストール不要)

The screenshot displays the Visual Studio Code interface with a Wireshark capture of network traffic. The terminal window shows the command `docker ps` and its output, listing several containers including `popcornsar_r20-11_wr1_lts21:1.1.1` and `test`.

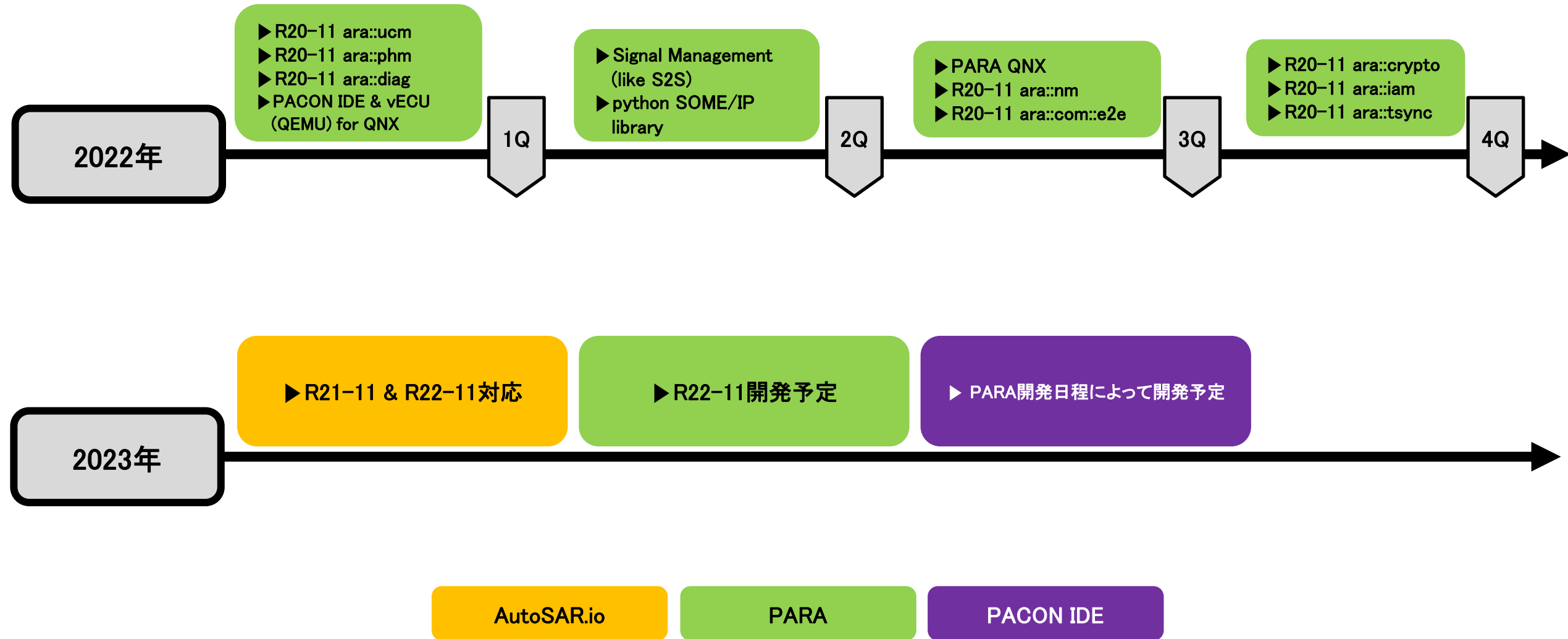
| No. | Time     | Source     | Destination | Protocol   | Length | Info  |
|-----|----------|------------|-------------|------------|--------|---|
| 1   | 0.000000 | 172.20.0.5 | 224.0.0.1   | SOME/IP-SD | 112    | SOME/IP Service Discovery Protocol [Offer]        |
| 2   | 1.632546 | 172.20.0.4 | 224.0.0.1   | 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]        |

## 6. 当社から構築するAP開発環境のプロセス

- QEMUとYoctoは使いません。
- APバージョンによりARA APIが変更され、これによって独自のテストSWを開発します。



## 7. 当社のAP開発ロードマップ(2023年まで)





## (補足資料1)製品デモーの動画

---

1. R20-11 Autosar.io ARXML modeling :

– リンク : <https://youtu.be/3FYzR0bQ44s>

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

– リンク : <https://youtu.be/HYqNEMrYYAw>

3. R20-11 PopcornSAR's ARA::DIAG Demo

– リンク : <https://youtu.be/jvySoUdoAJs>

– リンク : [https://youtu.be/tm\\_Cr80d52w](https://youtu.be/tm_Cr80d52w)

# (補足資料2) 設計ツール(AutoSAR.io)のISO26262認定証



## 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               | ●                     |