



AUTOSAR Abstract Platform Description (or VFB++)

Introduction Presentation

Brian Rooney

INCOSE EMEA 2019

10 Oct 2019

Utrecht @ INCOSE EMEAWS

BMW
GROUP



DAIMLER



PSA
GROUPE

TOYOTA

VOLKSWAGEN
AKTIENGESELLSCHAFT

Agenda

> AUTOSAR Partnership

> Methodology

> Abstract Platform

> Design

> Validation

> Future Aspects

Agenda

> AUTOSAR Partnership

> Methodology

> Abstract Platform

> Design

> Validation

> Future Aspects

AUTOSAR Introduction



(AUTomotive Open System ARchitecture)

is a worldwide development partnership of car manufacturers, suppliers and other companies from the electronics, semiconductor and software industry.

The 280 AUTOSAR Partners



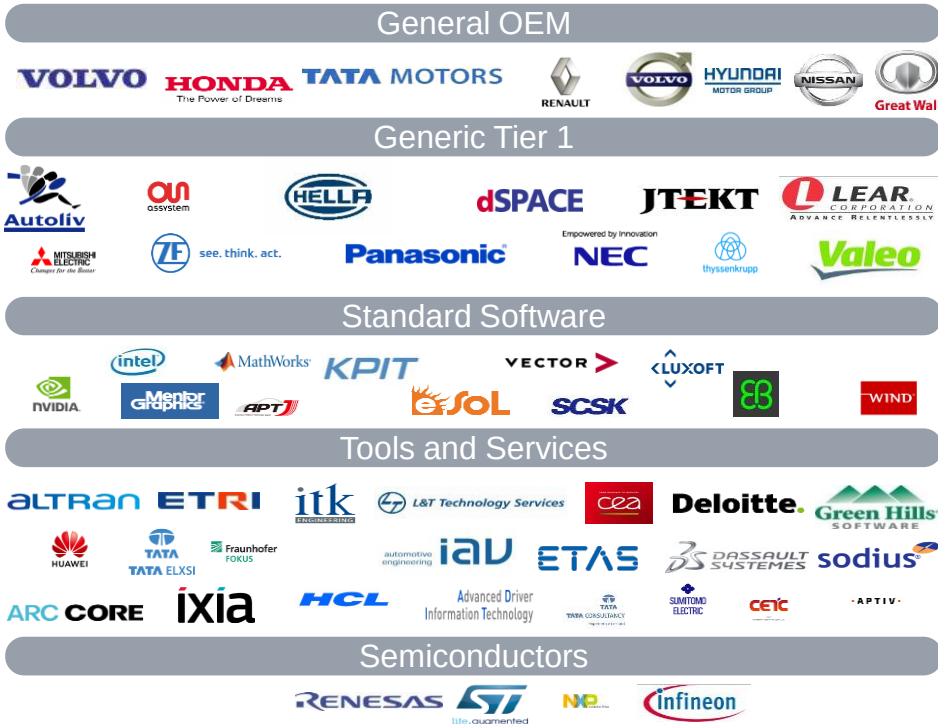
9 Core Partners



2 Strategic Partners



58 Premium Partners



49 Development Partners



134 Associate Partners
28 Attendees

AUTOSAR Partnership

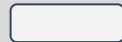
- AUTOSAR vision and goals
 - To standardize the software architecture of ECUs
 - Consolidate technical design elements into one standard
 - Hardware and software will be widely independent of each other
 - Flexible exchangeability of SW between OEMs and Suppliers
 - Platform defined layered architecture: applications – middleware - hardware
- AUTOSAR working groups
 - Cross Platform WGs: **Methodology**, **Architecture**, **Security**, **Safety**, IVC, V2X, Diagnostics, UCM
 - Platform specific WGs (incl. foundation):
 - Classic: RTE, MCB, MCAL, ...
 - Adaptive: EMO, PER, central coding team, demonstrator, ...

AUTOSAR Standards

Legend



Released as an own standard



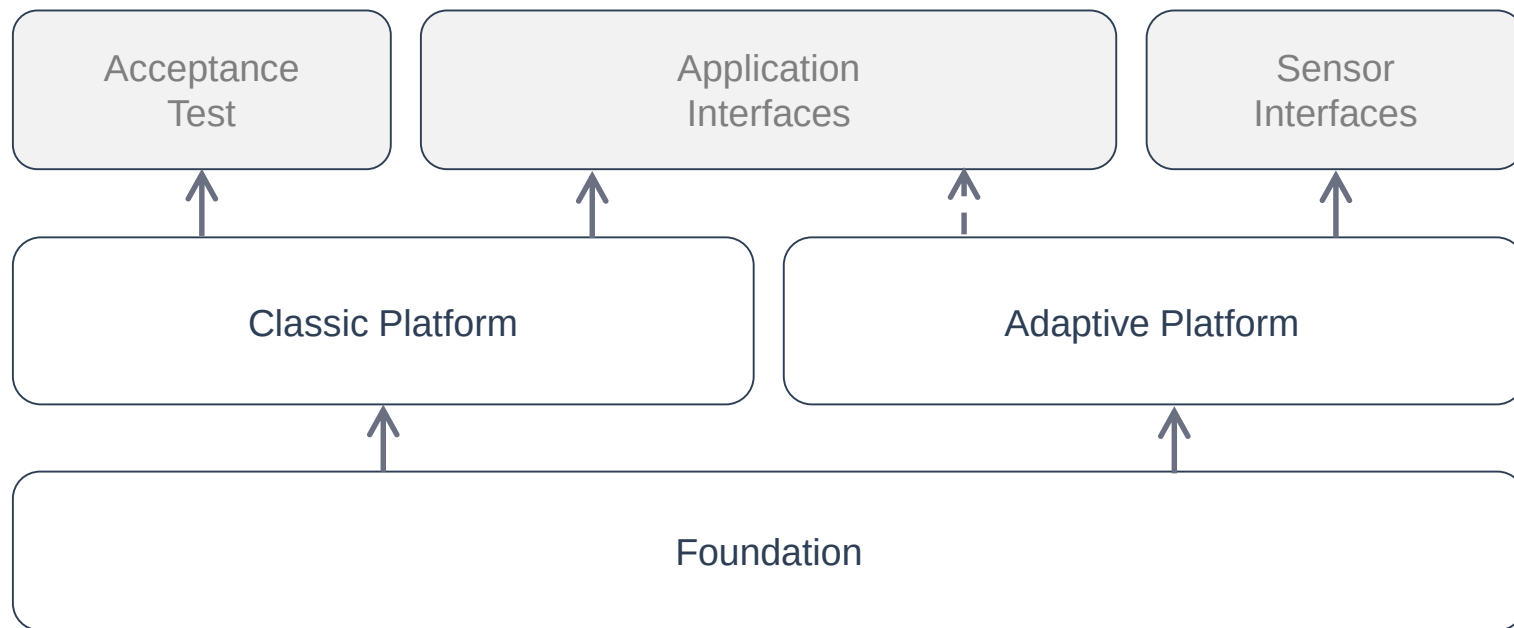
Released as part of the standard it is extending



A extends B

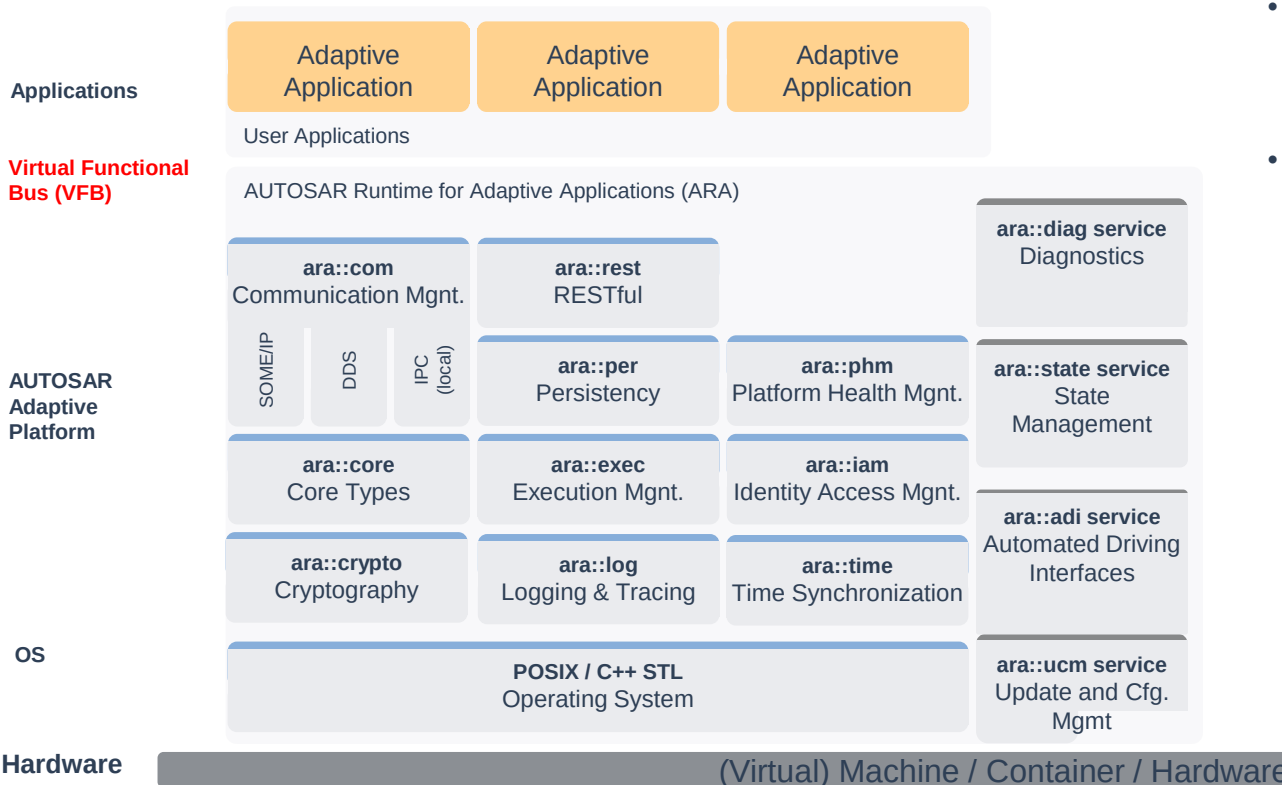


A planned to extend B



AUTOSAR Adaptive Platform

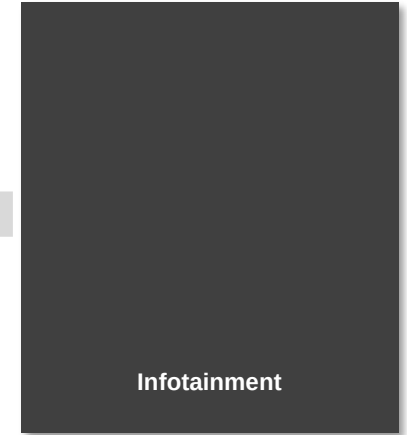
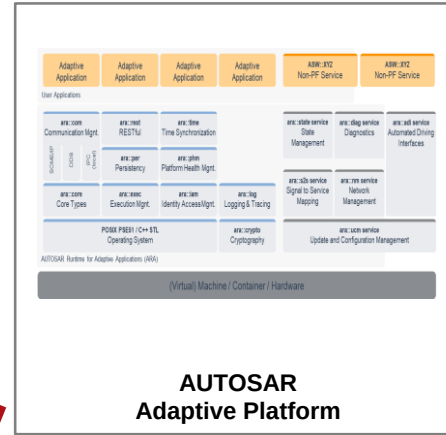
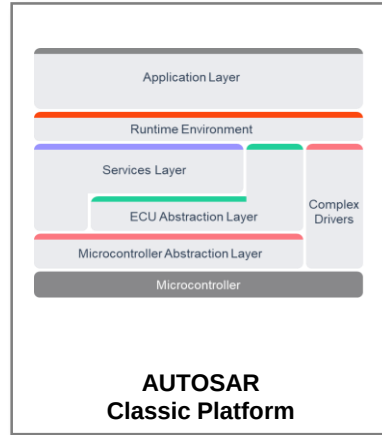
Logical view



- Lifecycle:
 - First planned “valid” release in November 2020
- Key Points of Adaptive Platform architecture:
 - Service oriented architecture (SOMEIP, DDS, REST)
 - Support for Virtual hardware (Hypervisor) deployments
 - Specifications are NOT hardware near
 - Targeting Non-time-critical scenarios (POSIX compliance)
 - No RTE, applications use direct interfaces to middleware modules (FCs)

AUTOSAR's answer to the upcoming challenges

platform of choice
for time-critical
and narrow-
bandwidth
applications



platform of choice for larger-bandwidth and less time-sensitive applications, for communication with other domains, e.g. infotainment, telematics, ...

③ BUT, in most cases, it's the OEM's decision which platform to target an application for – enter VFB++!!

Real time Requirements	High, in the range of micro-sec	Mid, in the range of milli-sec	Low, in the range of sec
Safety Criticality	High, up to ASIL-D	High, at least ASIL-B	Low, QM
Computing power	Low, ~ 1000 DMIPs	High, > 20.000 DMIPs	High, ~ 10.000 DMIPs

Agenda

> AUTOSAR Partnership

> Methodology

> Abstract Platform

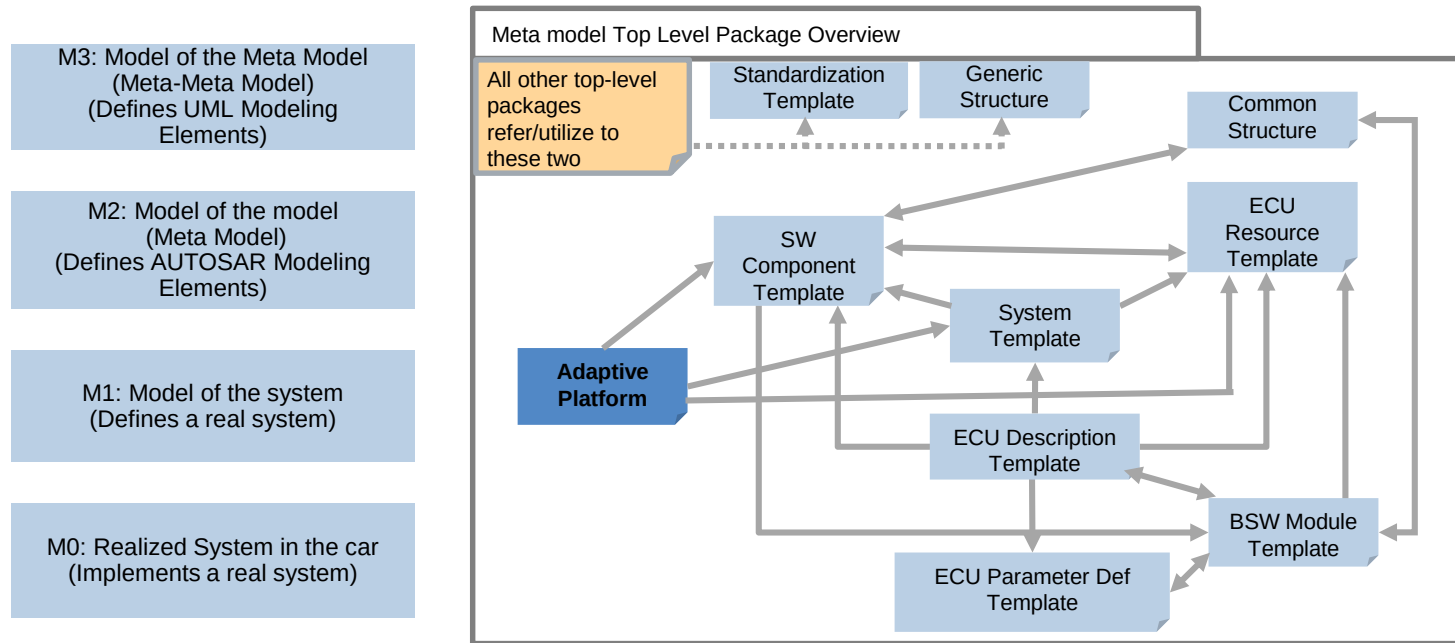
> Design

> Validation

> Future Aspects

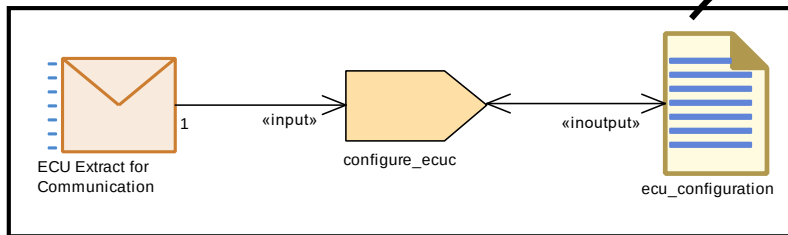
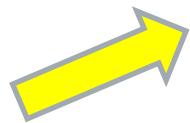
Methodology and the Meta Model

- The AUTOSAR Meta Model is the backbone of the AUTOSAR architecture definition
- Contains complete specification, how to model AUTOSAR systems
- Nowadays most changes occur in M2

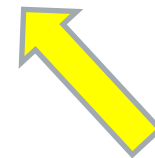
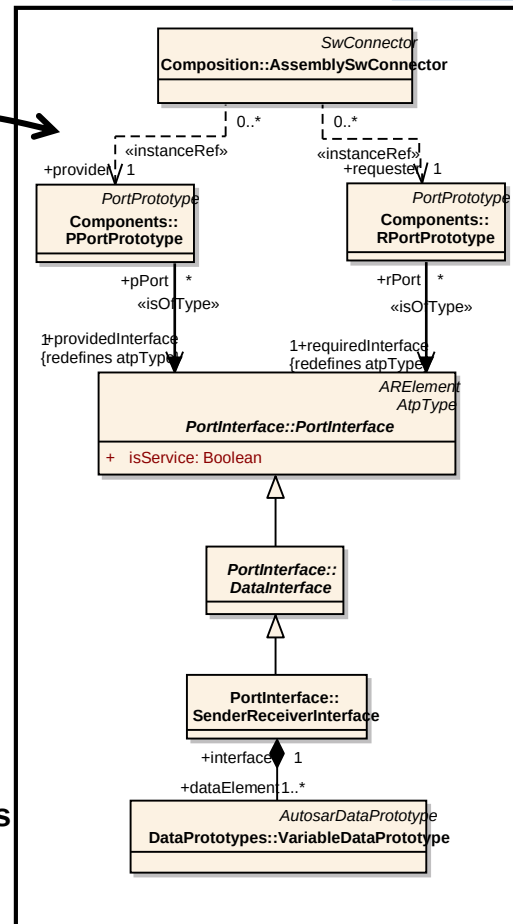


Methodology and the Meta Model

"How" and "what" to build in an AUTOSAR system



- **Methodology defines activities and work-products**
- **Meta Model defines the content:**
 - it is a formal description of all the information produced or consumed in the AUTOSAR methodology.
 - Is UML2 Based model (Enterprise Architect)
- **Meta Model is serialized to XSD Schema**
- **AUTOSAR exchange format for all model content is AUTOSAR XML (.ARXML)**



Meta Model
specifies
The "What"

Agenda

> AUTOSAR Partnership

> Methodology

> Abstract Platform

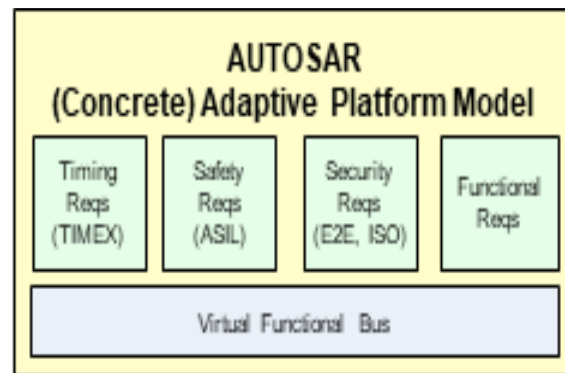
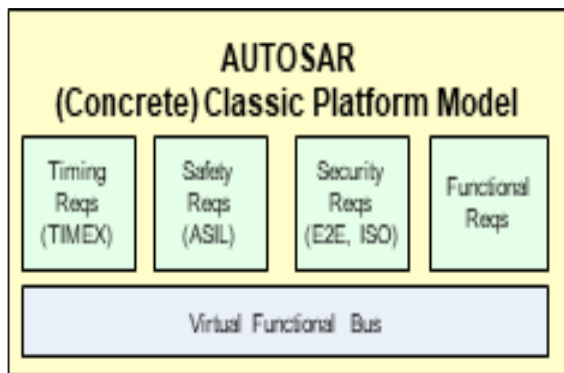
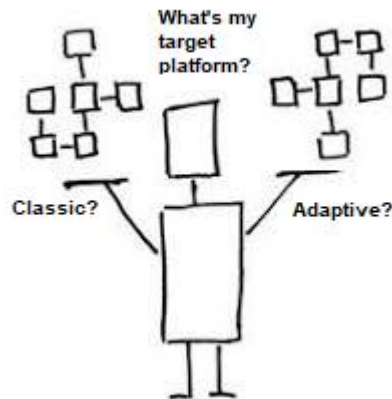
> Design

> Validation

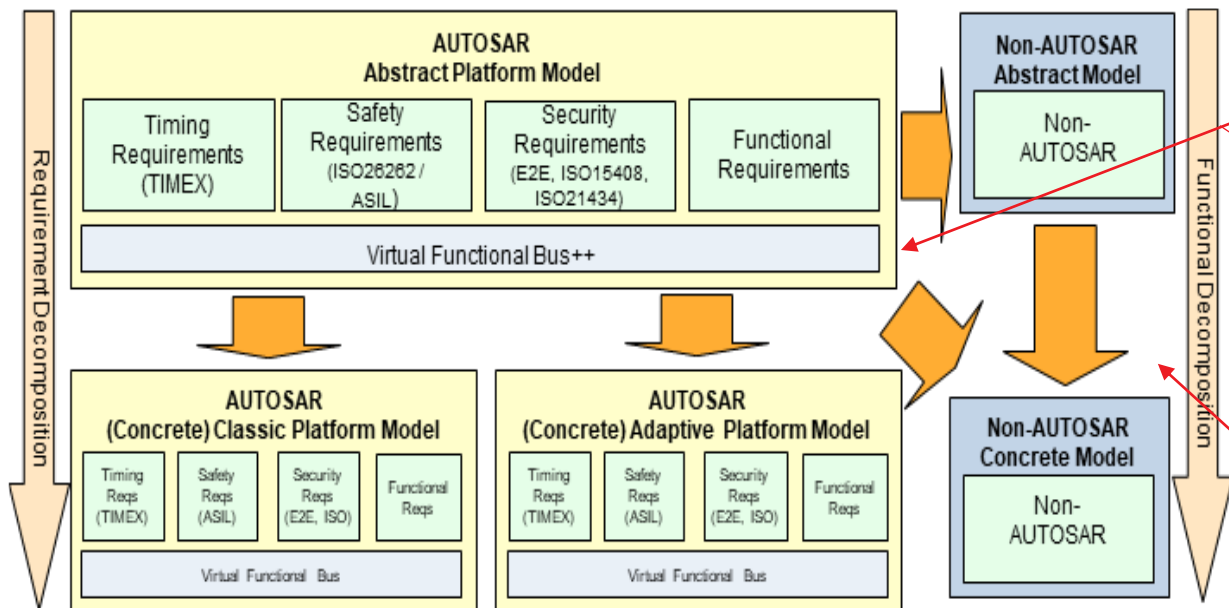
> ' Future Aspects

Background

- Existing concrete models are (intentionally) tightly coupled to the choice of platform
- A system designer is drawn a priori into a concrete decision whether to design and deploy on AUTOSAR adaptive platform or AUTOSAR classic platform or non-AUTOSAR platform = design bias!
- A system designer at an early software design stage may not necessarily care about what type of SW-C shall implement the function, or, which type of interface provides the required data
- The designer just wants to model the interaction between the functional software blocks: i.e. the signal name, the directional flow of the data (I/O) and the physical unit of the data on a high level and leave the further refinement of the design or indeed implementation details to a downstream stage, i.e. separation of concerns

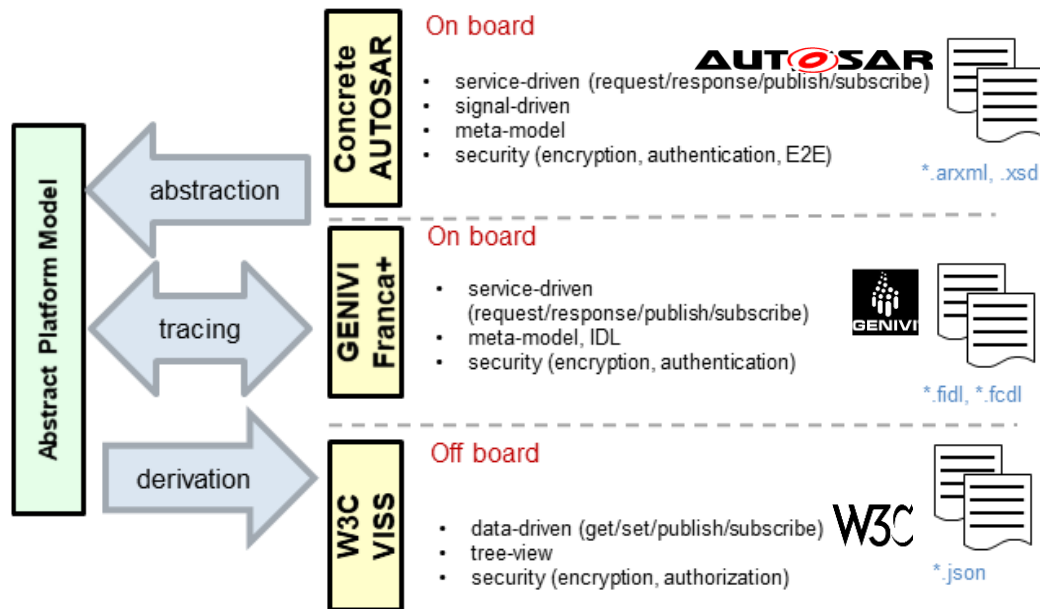


Motivation



- Platform independent description – no specifics about Classic or Adaptive
- VFB Level (SW Component Model) description
- Using existing AUTOSAR Meta-Model – quick wins on knowledge and tooling
- Potential usage with non-AUTOSAR (foreign) models
- Requirement decomposition and tracing
- Higher level view of the car communications matrix

Motivation



- Principally targeting AUTOSAR but should be useful to non-AUTOSAR e.g. GENIVI Franca+ or W3C VISS
- AUTOSAR Release 19-11
 - First draft targeting usage with Classic/Adaptive Platforms
- AUTOSAR Release 20-11
 - Bring in Franca+ or VISS aspects
 - Abstract – AUTOSAR Concrete Mapping

- Derivation: Derive the AUTOSAR concrete platform (VFB level) description from an abstract platform description
- Abstraction: It should be possible to create an abstract platform description from an AUTOSAR concrete platform description
- Tracing: requirement and functional tracing between the abstract and concrete models

Agenda

> AUTOSAR Partnership

> Methodology

> Abstract Platform

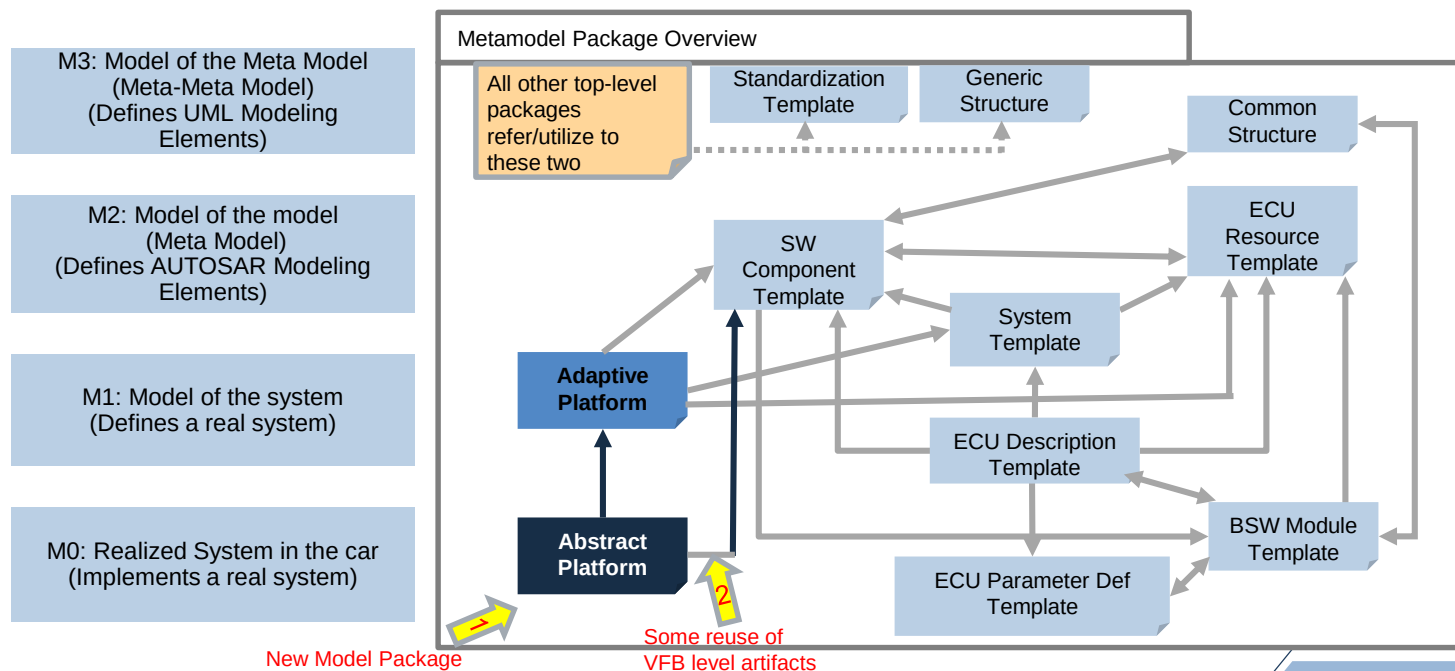
> Design

> Validation

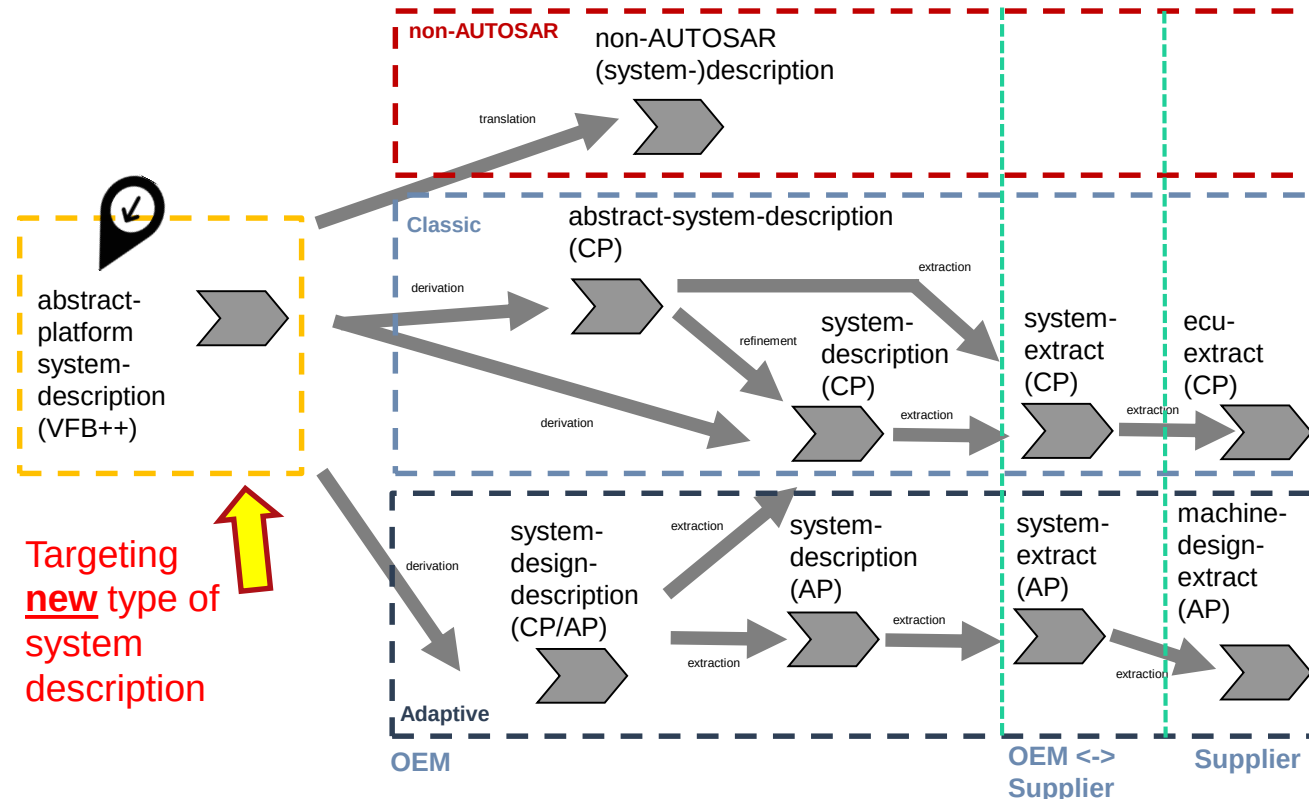
> Future Aspects

Meta Model Changes

1. New Abstract Platform top-level package
2. Since we are interested in modeling VFB level entities, we will need to reference model artifacts from SW Component Template



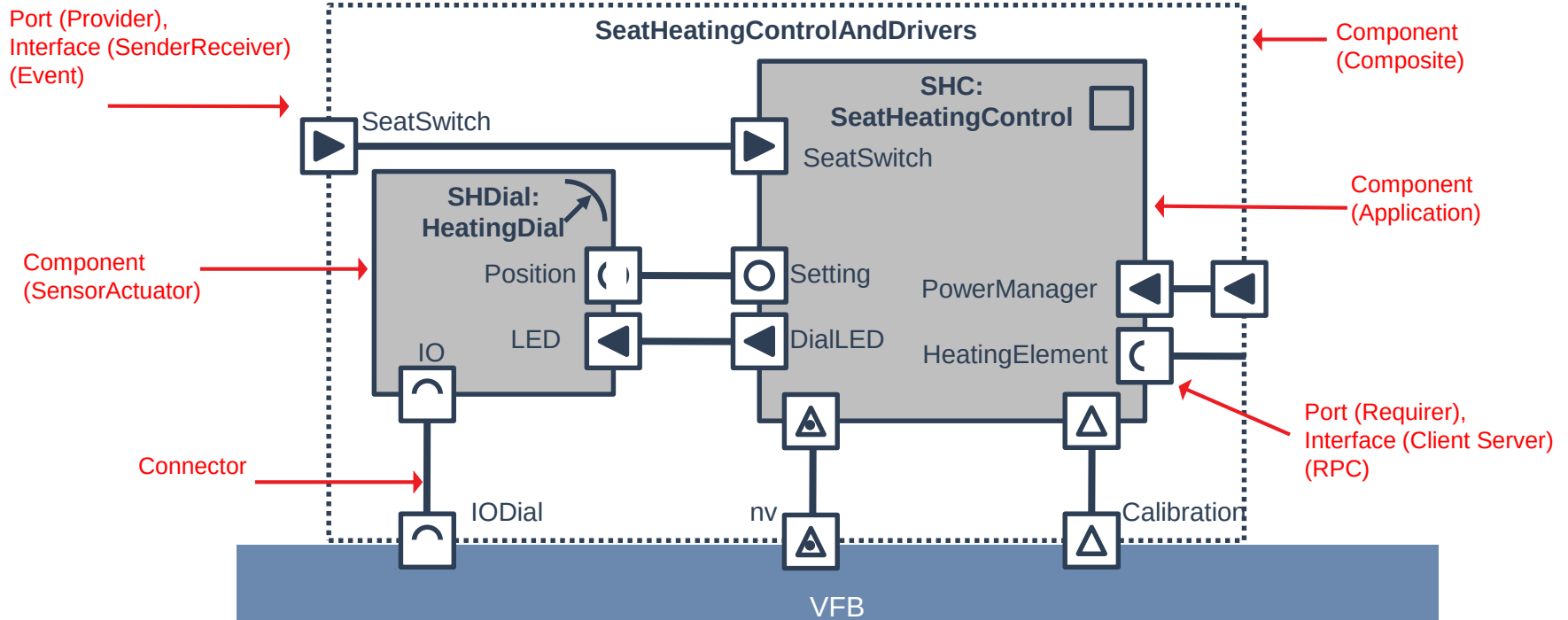
System Description Types



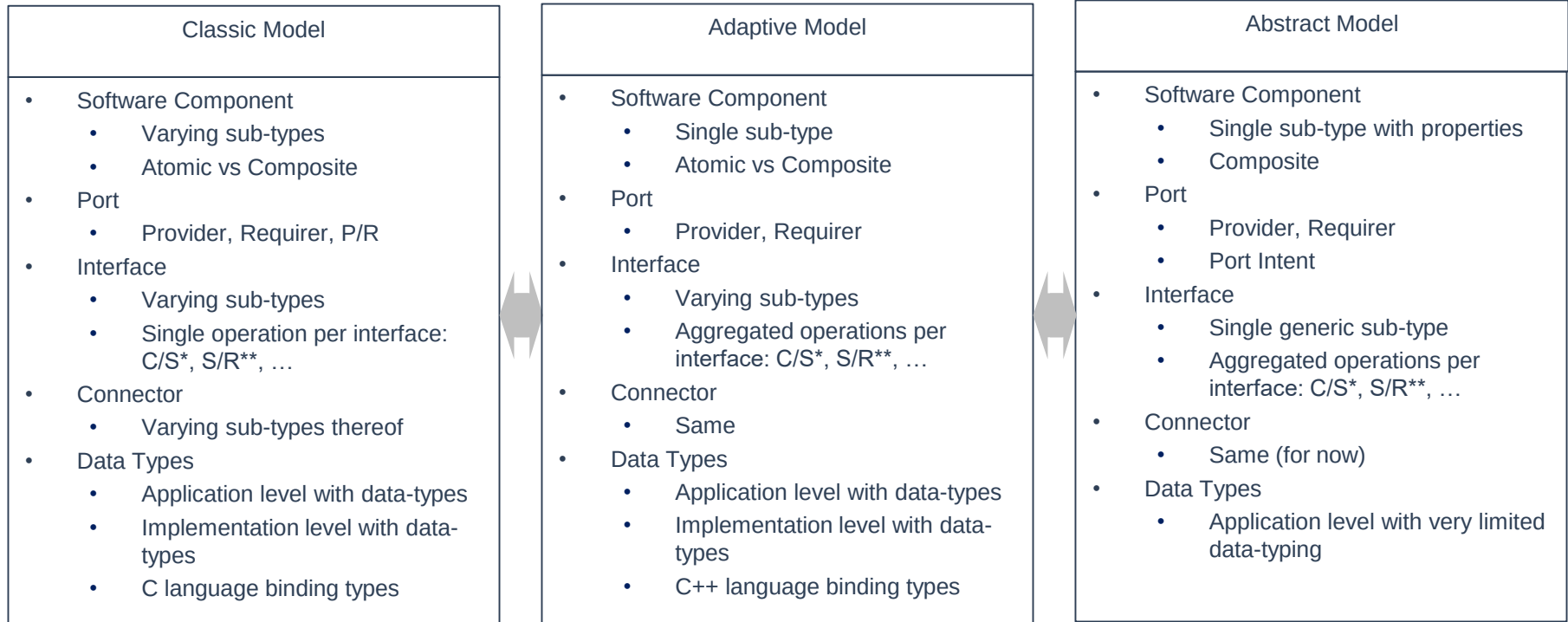
- Methodology should not forbid a system designer bypassing an abstract platform model and designing only in a concrete platform model (workflow backwards compatibility)
- Existing AUTOSAR system descriptions should not be functionally affected
- **Tooling is essential at all stages to make the transformations & configurations automatic!!**

VFB Level Model Elements

Graphical Example



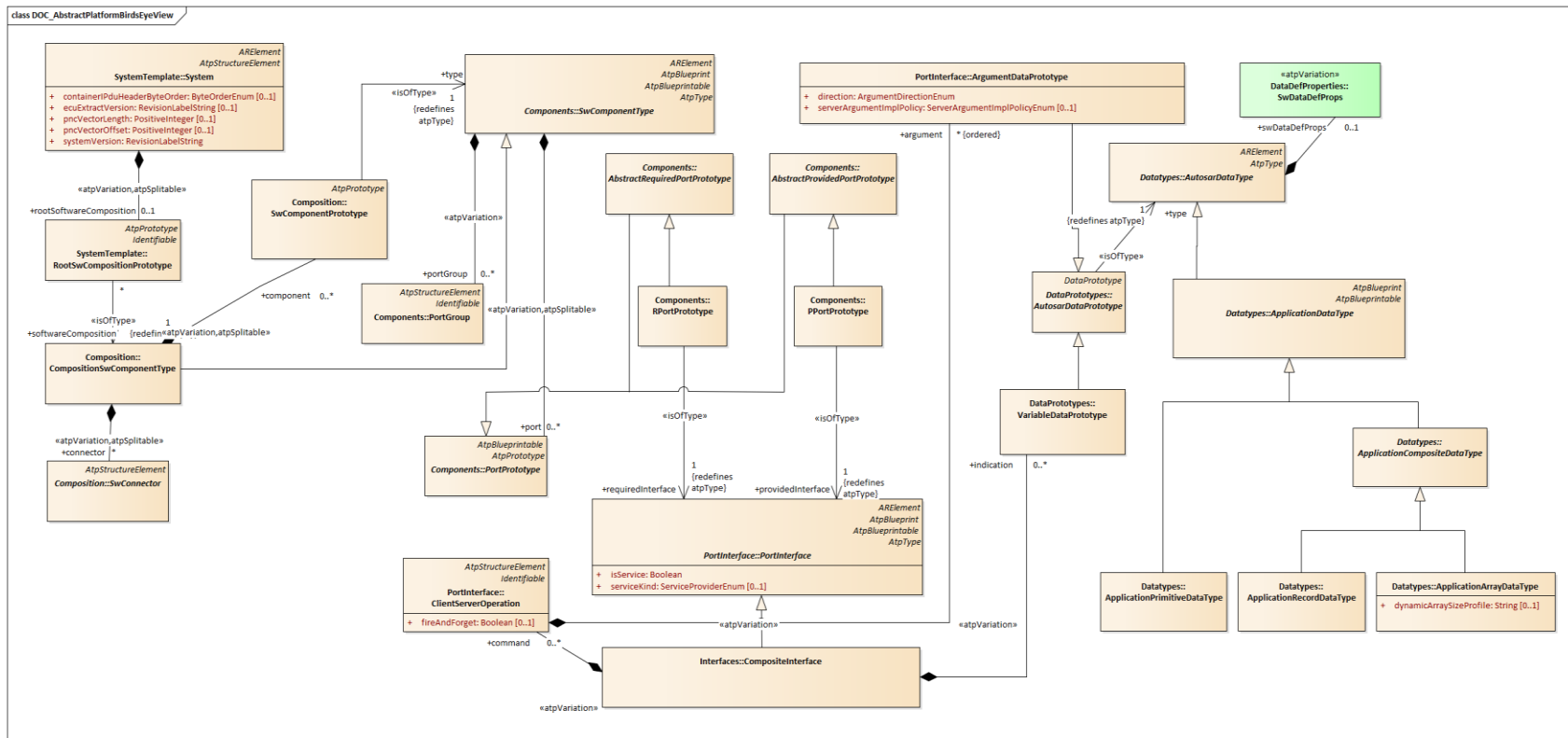
VFB vs VFB++ Level Model Elements Comparison



* C/S = ClientServer : a remote procedure call

** S/R = SenderReceiver : asynch. events

Abstract Platform Model



The diagram illustrates the relationships between four domains: system, components, ports, and data types. The diagram is organized into four main colored regions, each representing a domain, with various classes, interfaces, and data types connected by relationships.

- System Domain (Blue):** Contains classes like `SystemTemplate::System`, `AtpPrototype`, `Composition::CompositionSwComponentType`, and `Composition::CompositionSwComponentType`. It also includes a `SystemTemplate::System` class with attributes like `containerIdHeaderByteOrder`, `ecuExtractVersion`, `pncVectorLength`, `pncVectorOffset`, and `systemVersion`.
- Components Domain (Orange):** Contains classes like `ARElement`, `AtpBlueprint`, `AtpType`, `AbstractRequiredPortPrototype`, `AbstractProvidedPortPrototype`, `PortPrototype`, and `PortGroup`. It also includes a `PortInterface::ArgumentDataP` class with attributes like `direction`, `serverArgumentImpPolicy`, and `serverArgumentImpPolicyEnum`.
- Ports Domain (Yellow):** Contains classes like `PortInterface::PortInterface`, `ClientServerOperation`, and `Interfaces::CompositInterface`. It also includes a `PortInterface::PortInterface` class with attributes like `isService`, `serviceKind`, and `fireAndForget`.
- Data Types Domain (Green):** Contains classes like `DataTypes::AutosarDataType`, `DataTypes::ApplicationDataType`, `DataTypes::VariableDataPrototype`, `DataTypes::ApplicationCompositeDataType`, `DataTypes::ApplicationPrimitiveDataType`, `DataTypes::ApplicationRecordDataType`, and `DataTypes::ApplicationArrayDataType`. It also includes a `DataTypes::ApplicationCompositeDataType` class with attributes like `dynamicArraySizeProfile`.

The diagram shows various relationships between these domains, including inheritance, composition, and association. For example, the `System` domain is connected to the `Components` domain, which in turn is connected to the `Ports` domain. The `Ports` domain is connected to the `Data Types` domain, which is also connected to the `Interfaces` domain.

Agenda

> AUTOSAR Partnership

> Methodology

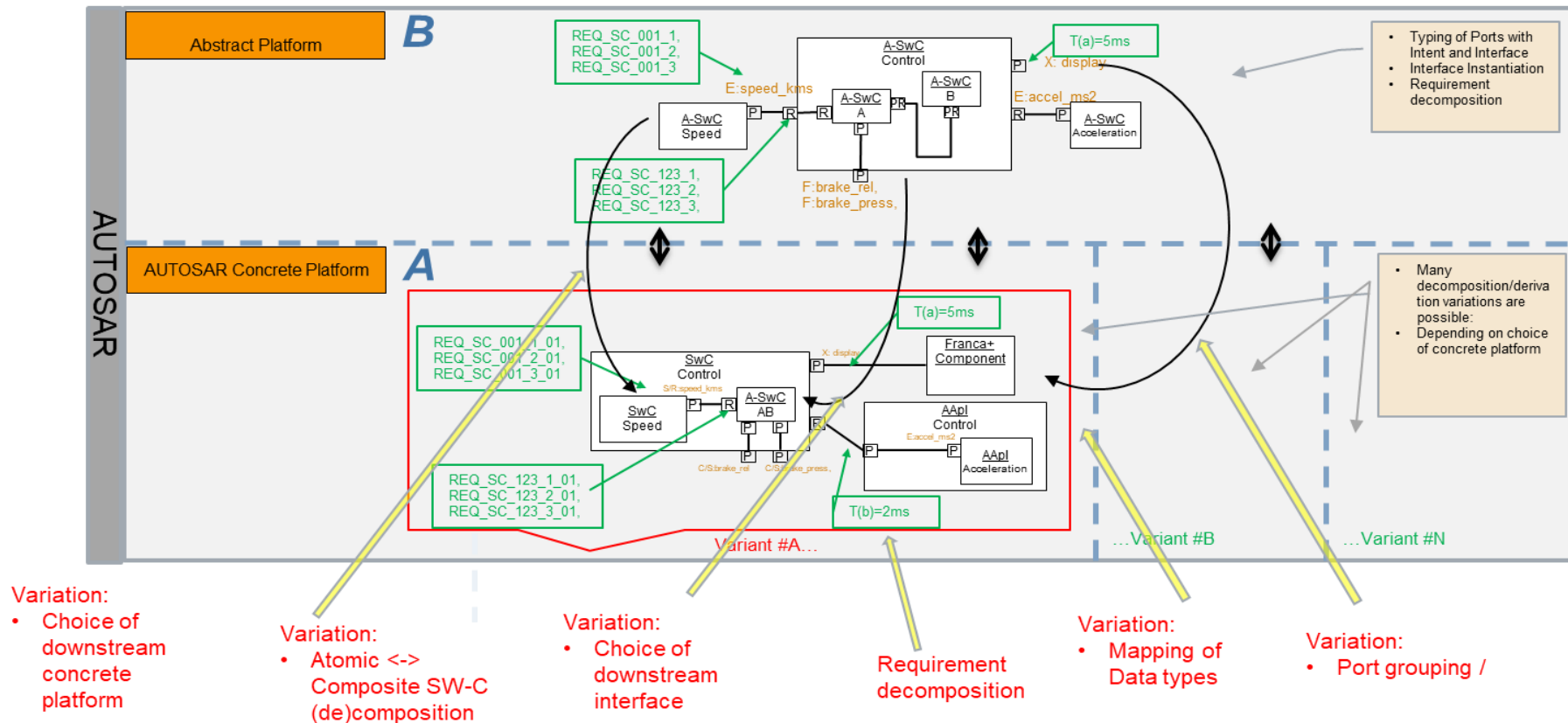
> Abstract Platform

> Design

> Validation

> Future Aspects

Validation Scenario



Agenda

> AUTOSAR Partnership

> Methodology

> Abstract Platform

> Design

> Validation

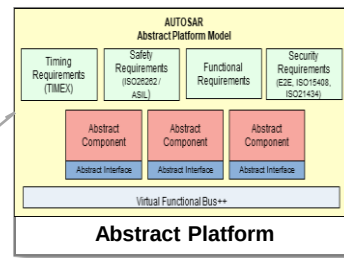
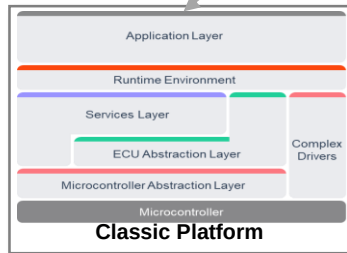
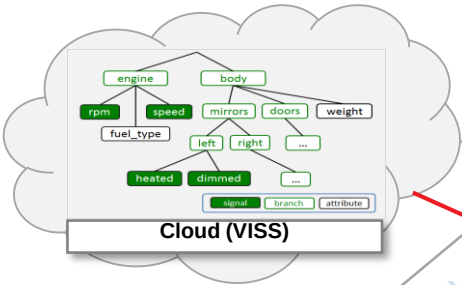
> Future Aspects

Future aspects

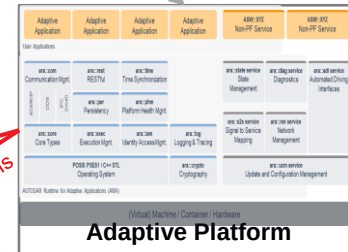
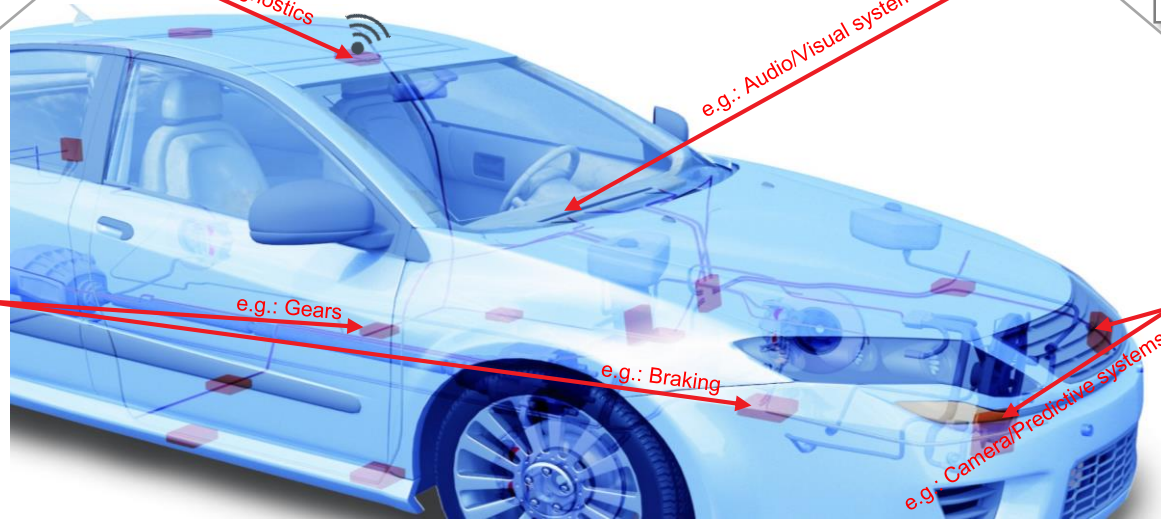
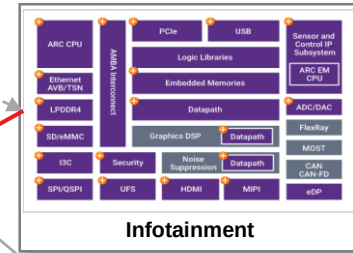
What's pending in AUTOSAR 20-11 and beyond?

- Define Abstract Platform Methodology
- Define mapping/guidelines between Abstract platform and AUTOSAR concrete platforms
- Define mapping/guidelines between Abstract platform and GENIVI Franca+
- Define mapping/guidelines between Abstract platform and W3C VISS
- Concept Validation

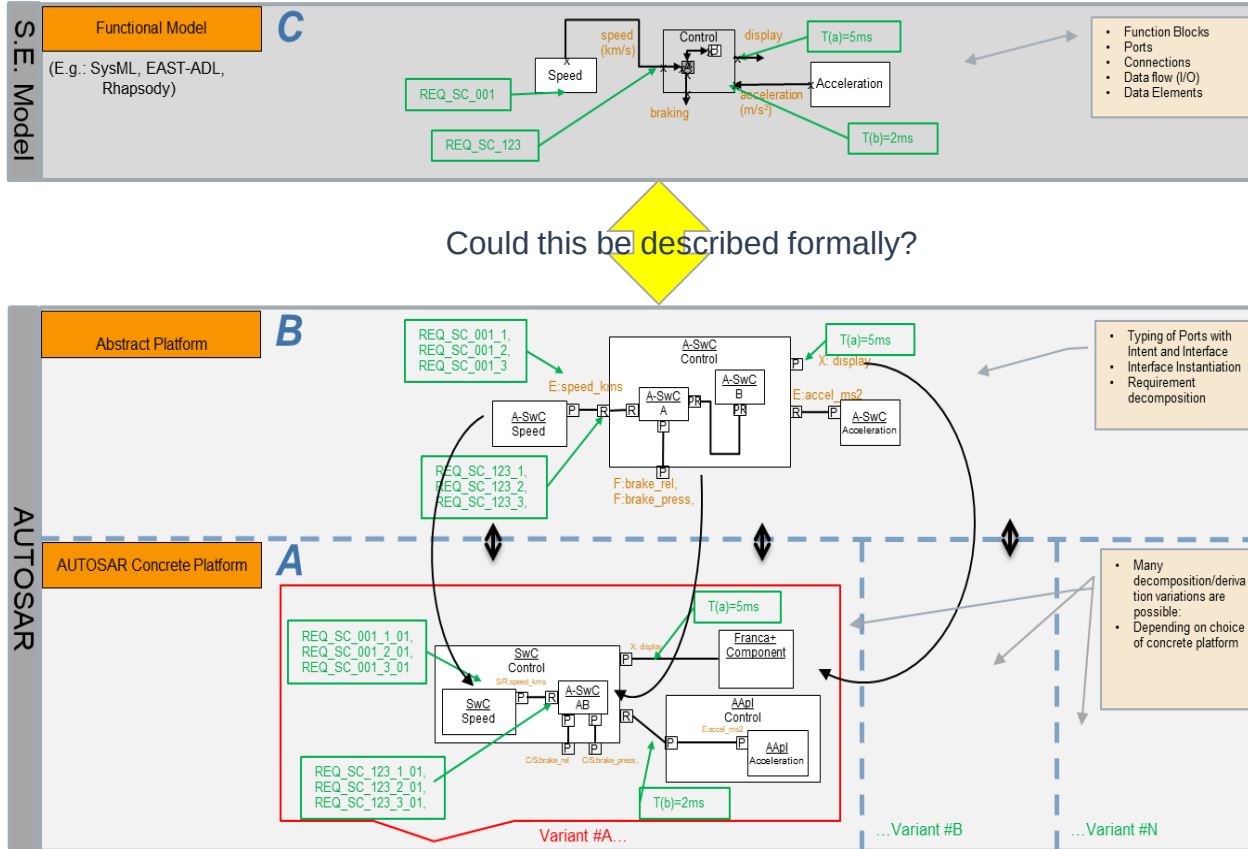
Future aspects



The vehicle will be a multi-standard/multi-platform mix of on-board and off-board systems - difficult to envisage a single super-standard taking the forefront... ..however, we can simplify how these platform models communicate.



Future aspects



- System engineering tools (e.g. SysML, EAST-ADL, Rational Rhapsody) and environments are upstream of this concept
- Such models are typically OEM specific
- It could be a use case to define a set of guidelines to describe how an S.E. model and an abstract platform model could “connect”

Exercise 1: Debug this code

```
#define/*__Int3rnti[n/l_()I3fuc^t3[]_C_C<>I7E_C[nt3st_*_/L/*MMXVIII_*/for#include/*'()'*)+,-./12357;:<=>?CEFGHIJKLMNOPUVWXYZ[]^`cfhijklmnrstuvwxyz{|}*/<stdio.h>  
char*r,F[1<2Y12W:"~T/>3(|+G(>/zUhy;Jx+5wG<v>)u55t.?sIZrC]n.;m+:l+hKjWjNji/Sh+2f1>c2H`)(_2(^L\ -  
|=([1/Z<2Y7/X12W:.VFFU1,T77S+N;M/>L..Kl+JCCI<H:(G*5F-Ee11C=5?.(>+(=3)Z-;*(:.Y/5(=-)2*-U,\ /+-  
?5('.,++++**'EEET,215IEUF.N`2':?GK;+^'^?>)5?>U>_)5GxG_.2K.2;);} _235[:5,S7E1(vTSS,-SSTvU(<-HG\ -2E2/2L2/EE-  
>E:?EE,2XMMMM1Hy')5rHK;+.T+?[n2/_2[LKN2/_|cK2+.2'];)?{KL57?|cK:2(NrHKtMMMk2nrH;rH[n" "Ckm_E21-E,-1->E(_ :mSE/LhLE/mm:2U1;2M>,2KW-  
+. -u).5Lm?EM`2`2nzXjj?[n<YcK?2]yC]H[^7N7LX^7N7UN</:-\ ZWXI<^I2K?>T?KH~-?fc;G_x2;;2XT7LXIuuVF2X(G(GVV--:  
:KjJ)HKLyN7UjJ3.WXjNI2KN<l|cKt2~[IsHfI2w{(<VV"  
"GifZG>x##&#&$&$;ZXIC####$&$#>7[LMv{&&#&#&#L,12TY.&#$#&#&$,(iiii,&&#&$#?TY2.$#s1(x###;2EE[t,\ SSEz.SW-k,T&jcQE-.## #&#57+$&$  
&#W1-&#s7W -J$#sKEN&s &#C^++$#W,h####n+/L2YE""2njK/H;YNs#s[,TU(#$, :&&~H>&# Y; &&G_x#2; ,mT&$YE-#& 5G  
$#VVF$#&zNs$&$Ej]HELy\ CN/U^Jk71<(#{G:7E+^&# l|?1 $$Y.2$ $ 7lz WzZw>&$E -<V-wE(2$ $ G>x; 2zsW/$$#HKt&$&$v>+tl(>"  
"7>S7S;,TT,&$;S7S>7&#E_-:U $ $","op ,*G= F,*I=957+F ;int*t,k,O, i, j,T+060<<+020];int M( int m,int nop){;;;return+ m%(0+nop );}  
int*tOo,w, h,z,W;void(C) (int n){n=putchar(n);}int f,c,H=11,Y=64<<2,Z,pq,X ;void(E/*d */) ( int/*RP*/n ){L(Z=k+00; Z;  
Z/=+2+000)G[000]=G*!!f |M(n,2)<<f,pq=2,f+=+06 <f?++pq,++pq,G++, z:f+001,n /=2;)}void (V) ( int/*opqrstabd*/n){C(n  
%Y);;C(n/Y+00);};void J(I)(L(pq--,pq =j =0=-1+0;++ j<240,I[6+ (h +6+j/12/2*2+m(j/2,2))* W+m(j/2/2,+06)*2+w*014 +-00+m(00+ 000+j,002  
+00)] =000 +00+k)k=M(G[j/2/2+( *-+  
32)**"<nopqabdeg"],/*4649&96#*/3);/*soaogogo*/; /*xD$P$Q$Rq*/int/*dbqpdpbpqxyzyzyboo35700Q*/main() {L(X=Y-  
1;i<21*3;i++,i++)L(r=G,G+=2;*G++;) *G>=13*3?*G-*r?*I++=*G:(*I++=r[l],*I++=r[2]):1;L(j=12,r =I;(*I=i=getchar())>-1;I++)i-7-3?I-  
=i<32||127<=i,j+=12:(H+=17+3,W=W<j?j:W,j=12);L(;*r>-1;r++)*r- 7-  
3?J(i),w++:(w=z,h+=17+3);C(71);C(73);V(' '*1*1*7);C(57);C(32*3+1);V(W);V(H);C(122*2);L(V(i=z);i  
<32*3);C(i++/3*X/31);C(33);C(X);C(11);L(G="SJYXHFUJ735";*G);C(*G++-5);C(3);V(1);L(V(j=z);j<21*3;  
j++){k=257;V(63777);V(k<<2);V(M(j,32)?11:511);V(z);C(22*2);V(i=f=z);V(z);V(W);V(H);V(l<<11);r=  
G=I+W*H;L(t=T;i<l<<21;i++)T[i]=i<Y?:-1;E(Y);L(i=-1;++i<W*H;t=T+Z*Y+Y)c=I[i]?I[i]*31-31:(31< j?j-31:31-  
j),Z=c[t[c]<z]E(Z),k<(l<<12)-2?t[c]=++k,T:t[t];E(Z);E(257);L(G++;k=G-r>X?X:G-r ,C(k),k);L(k--;C(*r++/*---  
#&$04689@ABDOPQRabdegoqp-----')));}C(53+6);return(z);}
```

...only joking 😊



Bem vindo a Lisboa!

12th AUTOSAR
Open Conference &
Networking Reception

12th AUTOSAR Open Conference



AUTOSAR Day

- Lecture- & Demo Sessions
- World Café
- Management Sessions

Tuesday,
March 3rd, 2020
9:00 am – 6:00 pm

Networking Reception

Tuesday,
March 3rd, 2020
7:00 pm – 10:00 pm

Conference

Wednesday,
March 4th, 2020
All-day



Venue:

NOVA School of Business & Economic
in Lisbon, Portugal

www.autosar.org/aoc2020