



NUTZEN DER STANDARDS xMCF UND FATXML IN DER CAE-PROZESSKETTE

AGENDA

1 Produktentwicklungsprozess – en gros und en détail

2 Datendurchgängigkeit – Generell

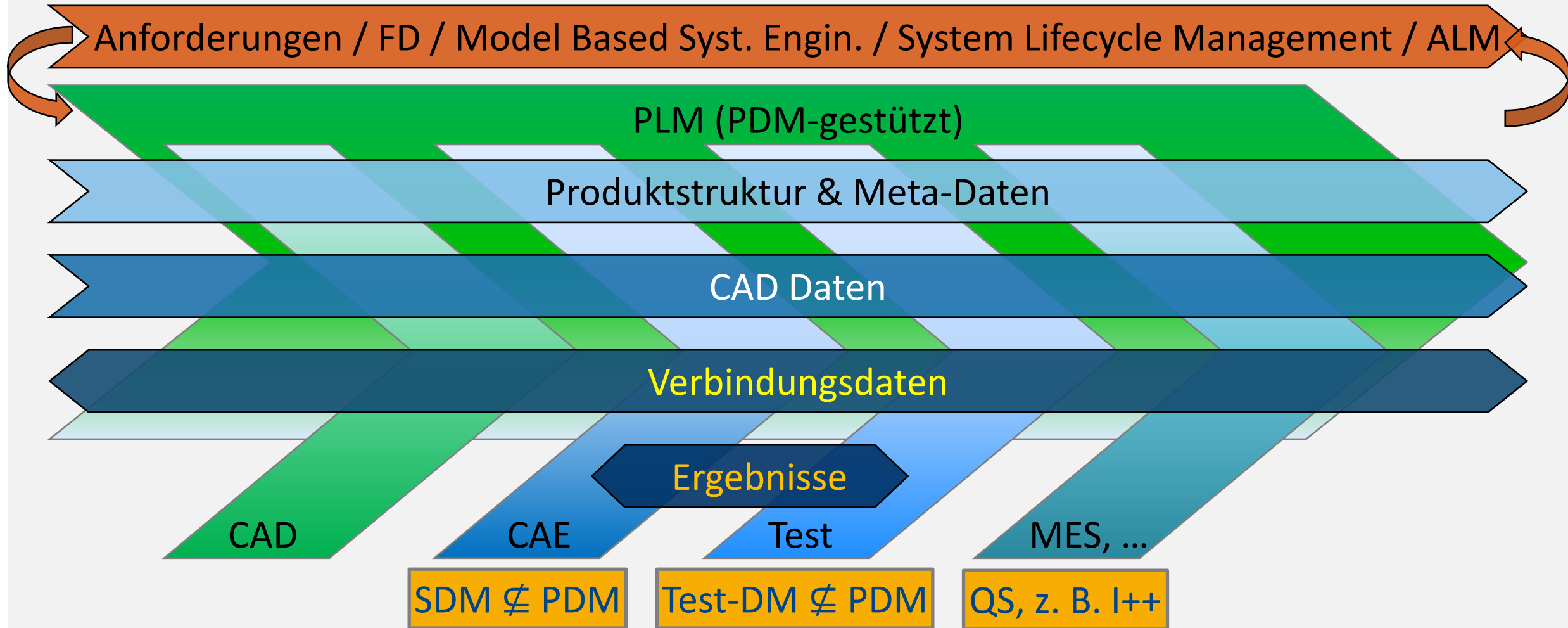
3 Bauteilverbindungen

4 Meta-Daten

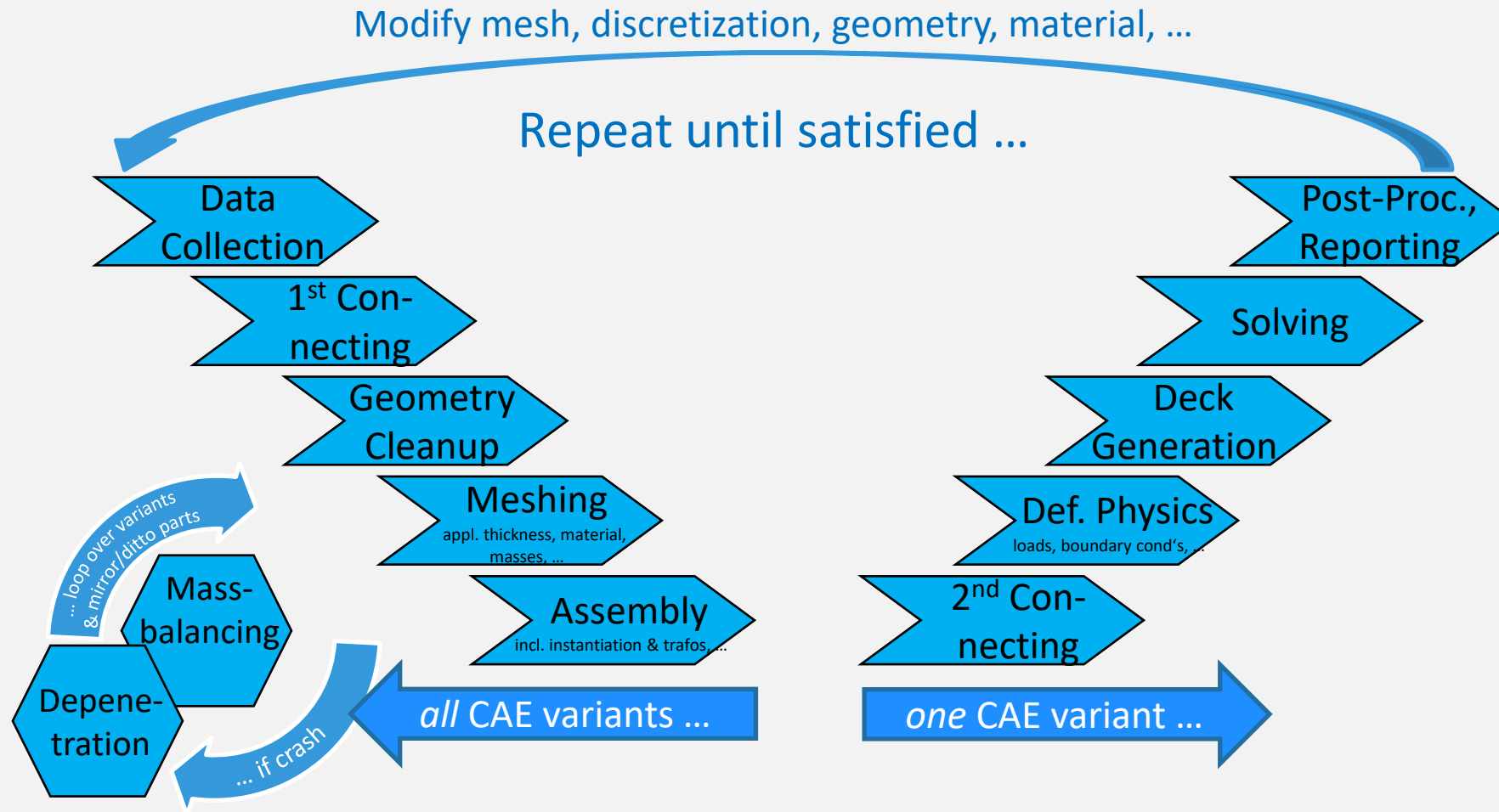
5 Synthese & Ausblick

PRODUKTENTWICKLUNGSPROZESS – EN GROS UND EN DÉTAIL

Wesentliche Domänen und Datenpfade im PLM

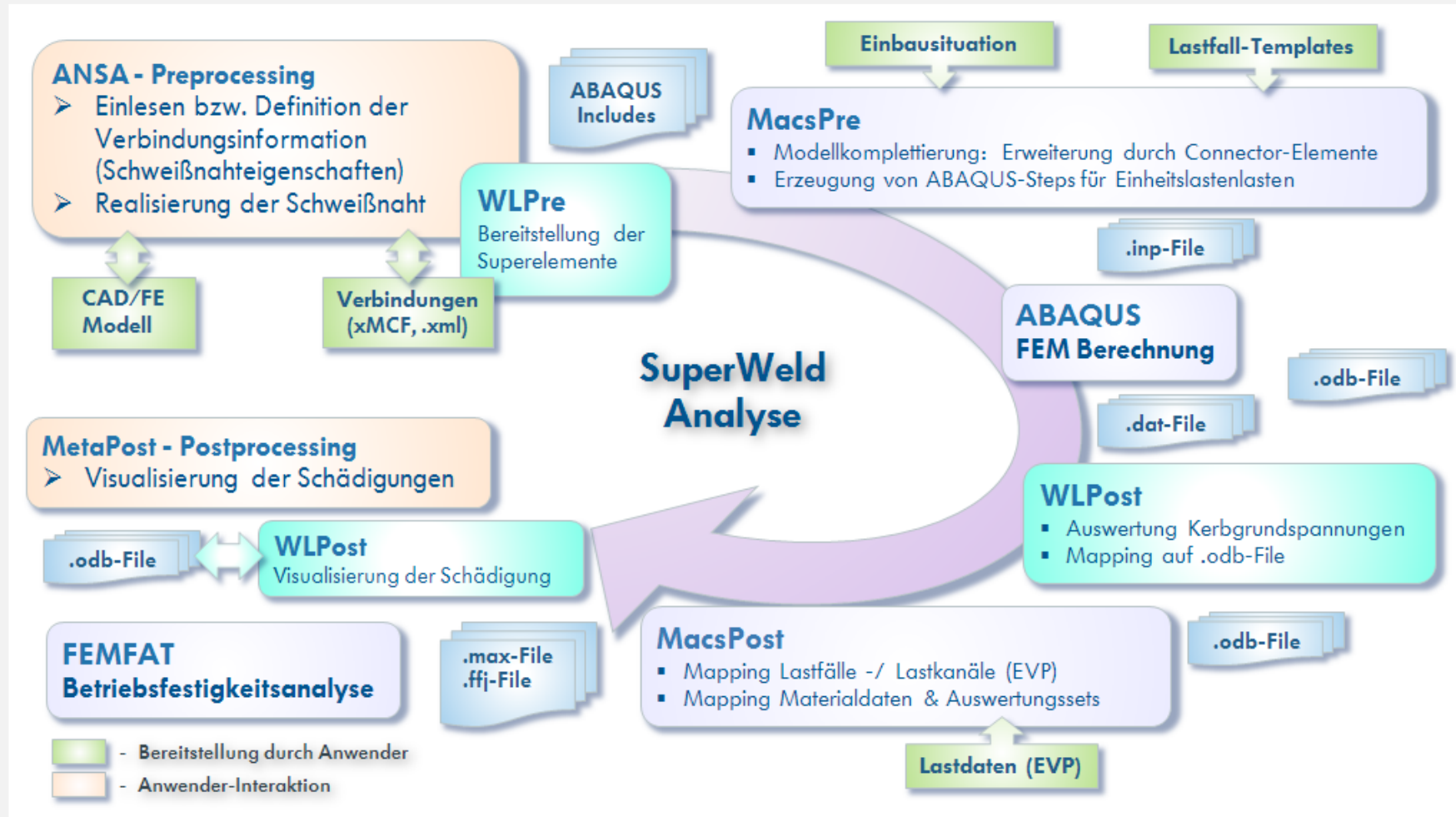


Ein typischer Simulationsprozess – *abstrakt* ...



... und konkret

Variante eines Fatigue-Prozesses bei Volkswagen





Fazit:

- Produktentwicklung ist auf allen Ebenen gekennzeichnet durch *zahlreiche Übergänge* zwischen Prozessschritten und zugehörigen Tools.
- *Datendurchgängigkeit* ist schwer zu erreichen.

DATENDURCHGÄNGIGKEIT – GENERELL

Anzahl und Komplexität von Schnittstellen

- Zwischen den Domänen (CAD, CAE, ...), innerhalb jeder Domäne und auch entlang einzelner Teilprozesse müssen hochkomplexe und umfangreiche Daten ausgetauscht werden.
- Komplexität und Umfang dieser Daten wachsen seit Jahren schnell. Das Wachstum wird sich beschleunigen.
- Die Anzahl für Automotive relevanter Software-Systeme hat sich *im Bereich CAD / PDM* im Lauf der letzten 2 bis 3 Dekaden konsolidiert:
 - Ca. jeweils ½ Dutzend Systeme teilen sich den Markt auf. In Deutschland Dassault & Siemens PLM.
 - Bei jedem OEM finden sich inzwischen ein zentrales PDM- und meist nur ein, maximal zwei CAD-Systeme.
- In den anderen Domänen, *ibes. für CAE, hat diese Konsolidierung nicht stattgefunden.*
 Zwar werden regelmäßig aufstrebende, kleinere Unternehmen von den großen Anbietern geschluckt.
 Die Branche und ihre Fragestellungen sind aber so innovativ, dass laufend neue kleine Unternehmen nachwachsen.
 Und auch „in house“ werden laufend neue eigene Tools entwickelt.
- ➔ Auch auf absehbare Zeit wird es noch notwendig sein, die o. g. Daten zwischen *vielen verschiedenen Software-Systemen* vieler verschiedener Hersteller auszutauschen.

Konsolidierung vs. Innovation

- Dassault Systèmes erwarb :
 - Abaqus: Einstieg in CAE, 2005; seit 2010 auch CFD
 - Dymola: Modelica, 2006
 - Simulayt: Composites, 2011
 - FE Design: Optimierung, 2013
 - CST: Elektromagnetismus, 2016
 - Xflow: Lattice-Boltzmann, 2016
 - ...
- Siemens PLM erwarb:
 - UGS/NX: Einstieg in CAD & CAE , 2007
 - Vistagy: Composites, Verbindungstechnik, 2011
 - LMS: Durability, Messtechnik, Modelica, 2012
 - CD-adapco: CFD, 2016
 - Mentor Graphics: Elektromagnetismus, 2016
 - ...

Aufkommende Innovationen (Beispiele):

- Isogeometric Analysis: CAE auf Original-NURBS
- Lattice-Boltzmann: Mesh-Free CFD
 - Solver PowerFlow, LaBS (Multilevel) , ...
- Topologie-Optimierung unter Fertigungsrandbedingungen (Guss): Volkswagen; Vortrag SIMVEC 2016
- ...



Fazit:

- Wir brauchen Standards!

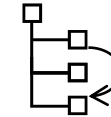
Standards müssen ...



... leicht erweiterbar sein bzgl. neuer Technologien und Parameter



... die schrittweise Ergänzung von Daten im Verlauf der Prozesse erlauben



... es Systemen ermöglichen, Daten einfach zu überspringen, die sie nicht benötigen



... sich in die Umgebung aller OEMs leicht einfügen lassen (PDM, Skripte, ...)



... zur Langzeitarchivierung geeignet sein

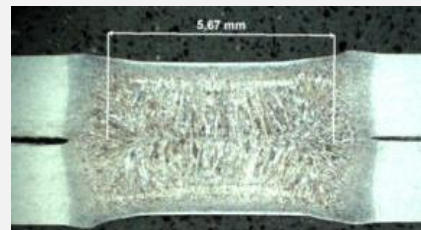
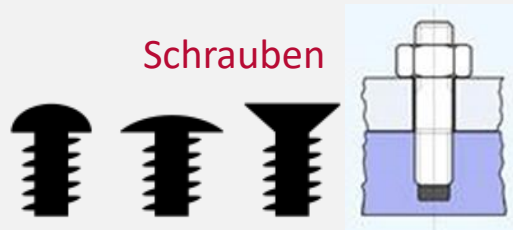


... idealerweise leicht menschenlesbar sein

➔ XML ist eine gute Wahl als Basis eines Standards.

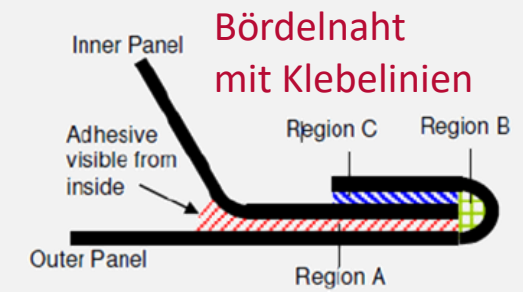
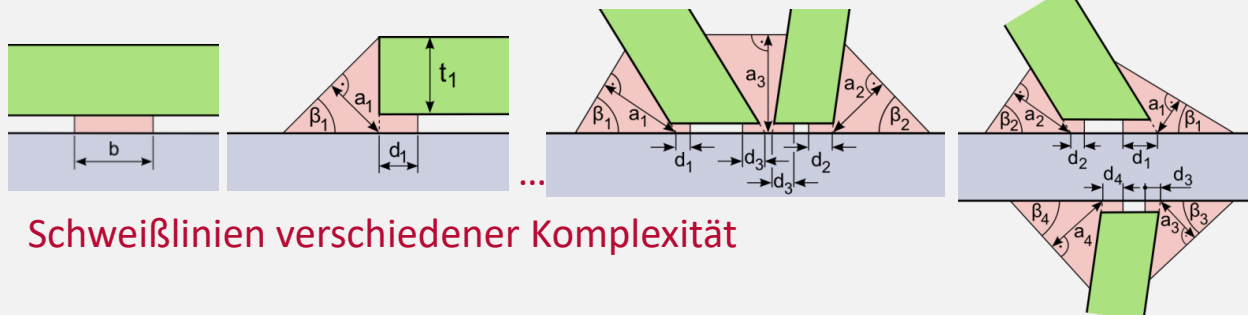
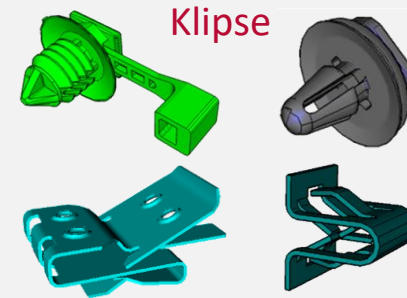
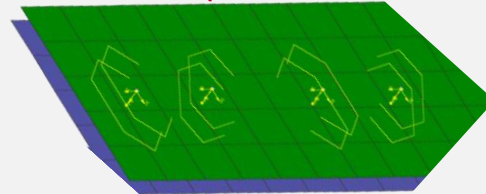
BAUTEILVERBINDUNGEN

Einige Füge-Elemente ...



Schweißpunkt (Schliffbild)

Robscans (im FE Pre-Prozessor)



Facts & Figures

- Einige 1.000 Schweißpunkte und Hunderte Schrauben pro Pkw
- > 100m Klebelinien im Pkw
- Etliche Meter Schweißlinien im Pkw (im Lkw noch wesentlich mehr)
- > 60 bekannte Fügetechnologien
- Bis zu 25 Qualitätskriterien pro Technik

Ursachen wachsender Technologie-Komplexität



Was ist das Besondere an Verbindungsdaten?

Verbindungsdaten unterscheiden sich von Produktstruktur, Meta- und CAD-Daten z. B. weil:

- ihre *Funktion* wichtiger ist als ihre Gestalt,
- sie mehr *Upstream-Datenfluss* als CAD-Daten verursachen,
- sie mit ganz anderen *Tools, Prozessen* und *Knowhow* erzeugt, verändert und optimiert werden,
- sie zu den *inneren Knoten* der Produktstruktur und nicht zu den Blättern gehören,
- ihr *Datenvolumen* bedeutend geringer ist, als das von CAD-Daten.

CAD- und Verbindungsdaten *ergänzen sich gegenseitig*. Das eine ist nutzlos ohne das andere.

Was ist das Problem? (Zumindest bisher ...)

- Jeder OEM setzt für Fügetechniken *selbst entwickelte* oder *proprietäre kommerzielle* CAD-Tools ein.
 - Gemeinsame *Entwicklungspartner* müssen „alles“ können.
 - Anbindung verschiedener Teilprozesse erfordert *viele zusätzliche Interface-Tools*, gerne auch „selbstgebraute“. Konsequenzen: *Hohe Kosten, Fehler, Aufwände*
 - Meist wird nur ein *Bruchteil der Fügetechniken* unterstützt – und nur ein *Bruchteil der Prozessparameter*.
 - Neue Prozessparameter oder gar Verbindungstechniken können *nur mit hohem Aufwand integriert* werden.
- ➔ Die Pflege all dieser Tools ist *teuer und fehlerträchtig*. Sehr leicht werden *etablierte Prozesse gefährdet*.
- Jeder Software-Wechsel erfordert *hohe Investitionen*.
- ➔ Der entsprechende “vendor lock-in-effect” *behindert den Wettbewerb und verzögert den Fortschritt*.



χMCF ist der Standard für Bauteilverbindungen

FAT | Forschungsvereinigung
Automobiltechnik **AK 25** | **Fügetechnik** definiert und pflegt χMCF.

χMCF erfüllt alle zuvor genannten Anforderungen, plus:

- Alle Verbindungsarten und Fügetechniken sind erfasst.
- Alle PLM-Teilprozesse werden unterstützt – CAD, CAE, CAT, CAM, inklusive spezieller Teilprozesse, z. B.:
 - Lebensdauersimulation,
 - Roboterprogrammierung,
 - Lieferantenanbindung, ...
- Eine χMCF-Datei enthält je nach Bedarf Daten einer *Baugruppe*, eines *Fahrzeugs* oder aller *Varianten* einer Baureihe.
➔ Jeder vorhandene Prozess kann χMCF verwenden.
- χMCF erlaubt spezifische Zusatzdaten von OEM / Prozess / Tool.

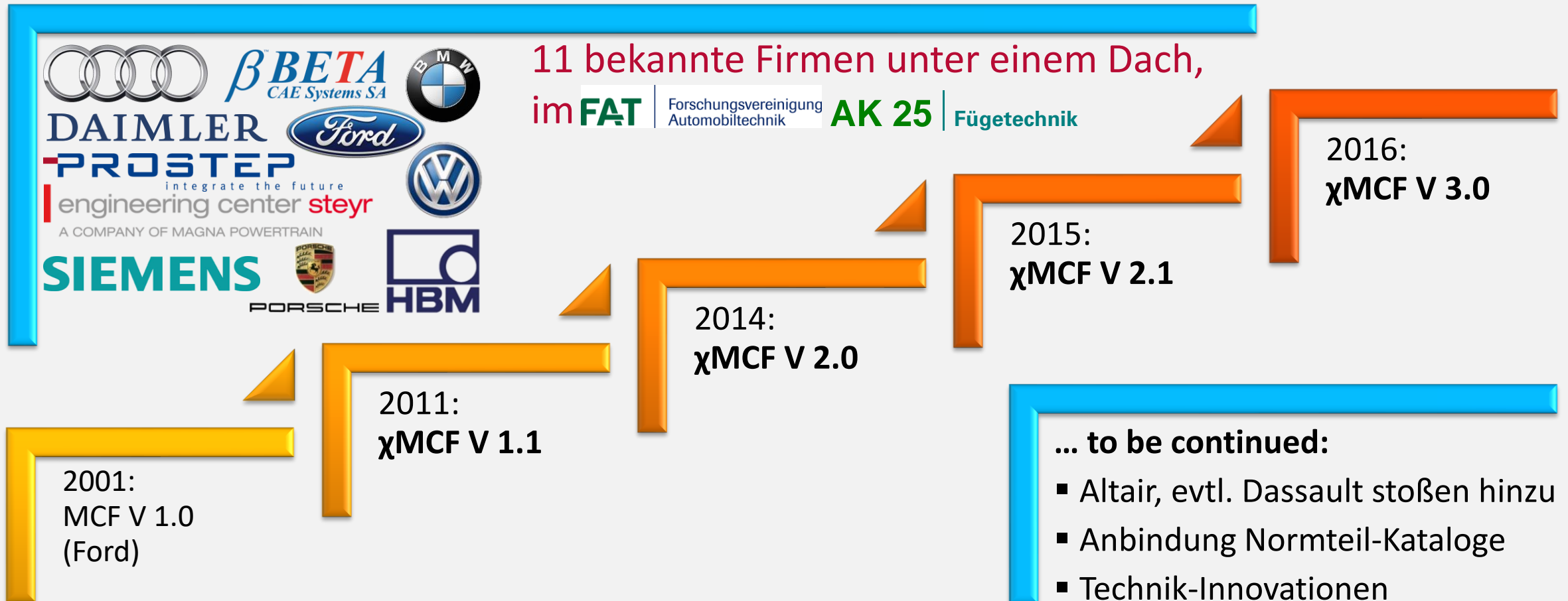
➔ Alle proprietären Formate können auf Dauer abgelöst werden.

Wachstum von χ MCF Daten entlang der Prozesskette **PROSTEP**



χMCF – Neuer Standard mit langer Geschichte

PROSTEP





Fazit:

- Fügetechniken werden immer komplexer.
- Verbindungsdaten müssen besser fließen.
- Dafür brauchen wir einen flexiblen Standard.
- χ MCF wurde für diesen Bedarf entwickelt.
- Es bildet die ideale Basis zur Prozess- und Tool-Integration.

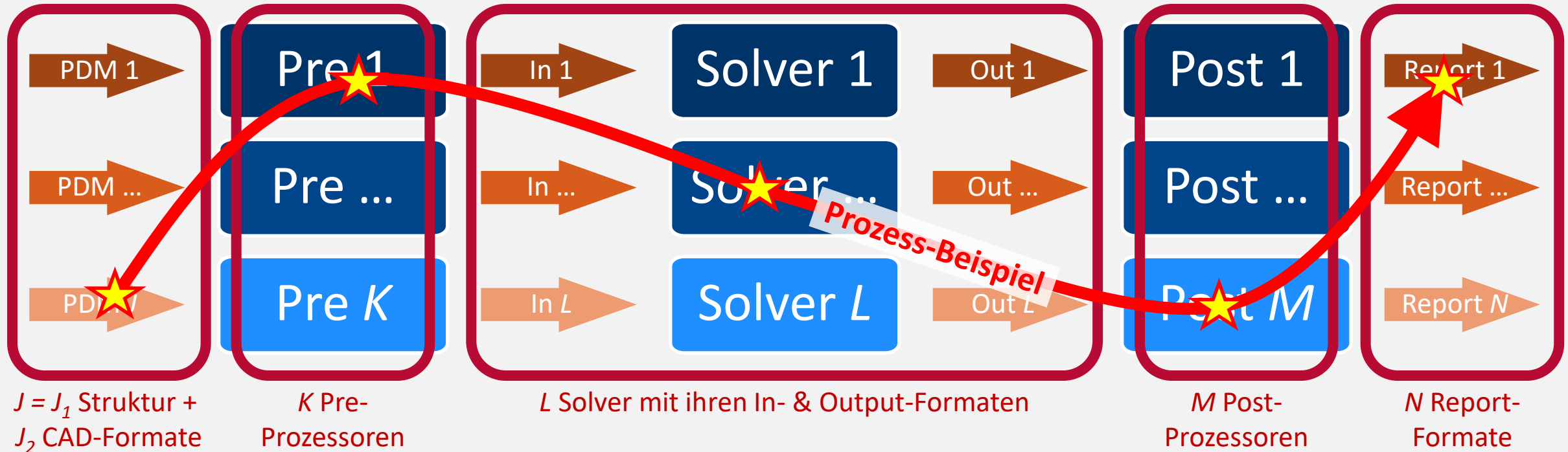
➔ χ MCF wird „Lingua franca“ für Fügedaten, genau wie es JT für CAD-Daten wurde.

META-DATEN

- Meta-Daten werden im PLM allgemein immer zahl- und bedeutungsreicher.
- Vieler dieser Meta-Daten sind auch für CAE-Prozesse bedeutungsvoll, z. B.:
 - Sachnummer, Teilename
 - Produktstruktur, Varianten, Konfigurationen, Gültigkeiten, Versionen
 - Ggf. abweichende CAE-Modellstruktur
 - Material, Wandstärke, Fertigungsparameter (KTL, ...)
 - Freigabestatus, Ansprechpartner (Bauteilverantwortlicher, Konstrukteur, ...), Änderungsdatum, ...
 - Masse & Trägheit – geschätzt, berechnet, gemessen
 - Farbe, Transparenz

Aber: Zu oft gehen Meta-Daten im CAE-Prozess verloren! *(vgl. nächste Folie)*

Datenflüsse im einstufigen* CAE-Prozess



Anforderung (Beispiel): *Der Report soll aus Post-Processing automatisch Versionsstand & CAE-Variante relevanter Teile übernehmen.*
Problem (Beispiel): *Wie kommen Versionsstand & CAE-Variante nach Post-Processing?*

*:= Nur ein Solver-Aufruf beteiligt, also z. B. *kein* voller Fatigue-Zyklus. Nur jeweils ein Pre- und Post-Processor beteiligt.

Meta-Daten in CAE-Prozessen

– Mindestanforderungen

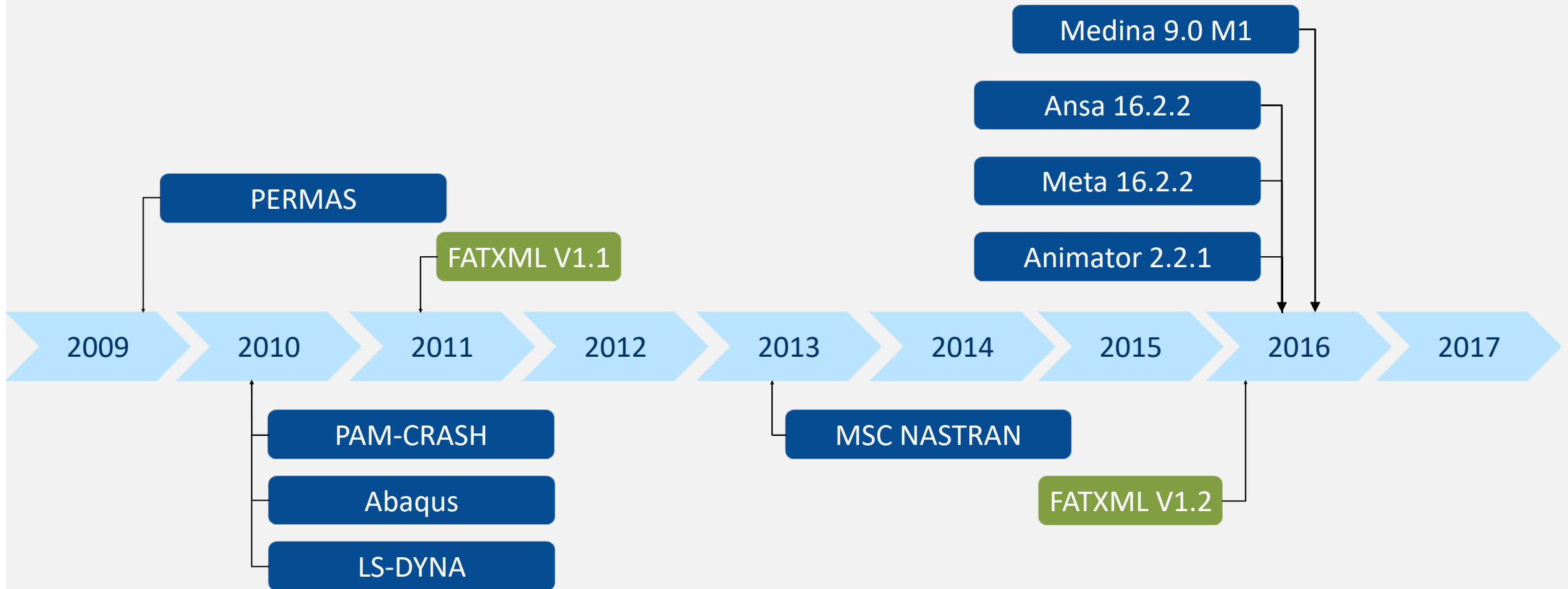
- Pre-Processoren müssen Modellstruktur inkl. Meta-Daten möglichst komplett aus allen Struktur- und CAD-Formaten sowie aus allen Solver-Input-Formaten in alle Solver-Input-Formate übertragen.
- Solver müssen diese Informationen unverändert und ohne Verlust in ihre Output-Formate übertragen.
 - Es kann sinnvoll sein, wenn Solver bestimmte Meta-Daten interpretieren (z. B. zwecks Modell-Check), oder wenn sie Meta-Daten ändern/ergänzen (z. B. berechnetes Gewicht, Konvergenzverhalten, ID-Offset, Include-Name) .
 - Dies ist aber eine weiterführende Anforderung.
- Post-Processoren müssen die Informationen aus allen Solver-Output-Formaten importieren & interpretieren (z. B. Darstellung Modellstruktur).
- Die Darstellung in Solver-Input-Formaten muss mit Include-Dateien verträglich sein.
- Existierende CAE-Prozesse dürfen nicht beeinträchtigt werden.
- *(weiterführend)* Die komplette Datenbasis des Pre-Processors soll sich aus dem Solver-Deck restaurieren lassen.

Die Vielzahl der möglichen Datenflüsse auch nur im einstufigen CAE-Prozess erzwingt eine systematische und einheitliche Herangehensweise.

Grundideen:

- Die zu einem (Include-)Deck gehörenden Meta-Daten werden genau *im jeweiligen Deck* angeordnet. Die Daten werden durch *spezielle Solver-Syntax* gekapselt und in den Output übertragen. (Anders als z. B. Kommentare)
 - Syntax-Restriktionen des Solvers (Zeilenlängen, Sonderzeichen) werden durch eine Kodierung nach einem allgemeinen, Solver-unabhängigen Verfahren berücksichtigt.
- Die Syntax zur Darstellung der Meta-Daten basiert auf XML.
- Die *XML-Syntax* ist generell *unabhängig vom Solver*.
- Lediglich die Modell-Objekte (Elemente, Properties, ...) werden in Solver-Syntax adressiert.

FATXML-Zeitschiene





Fazit:

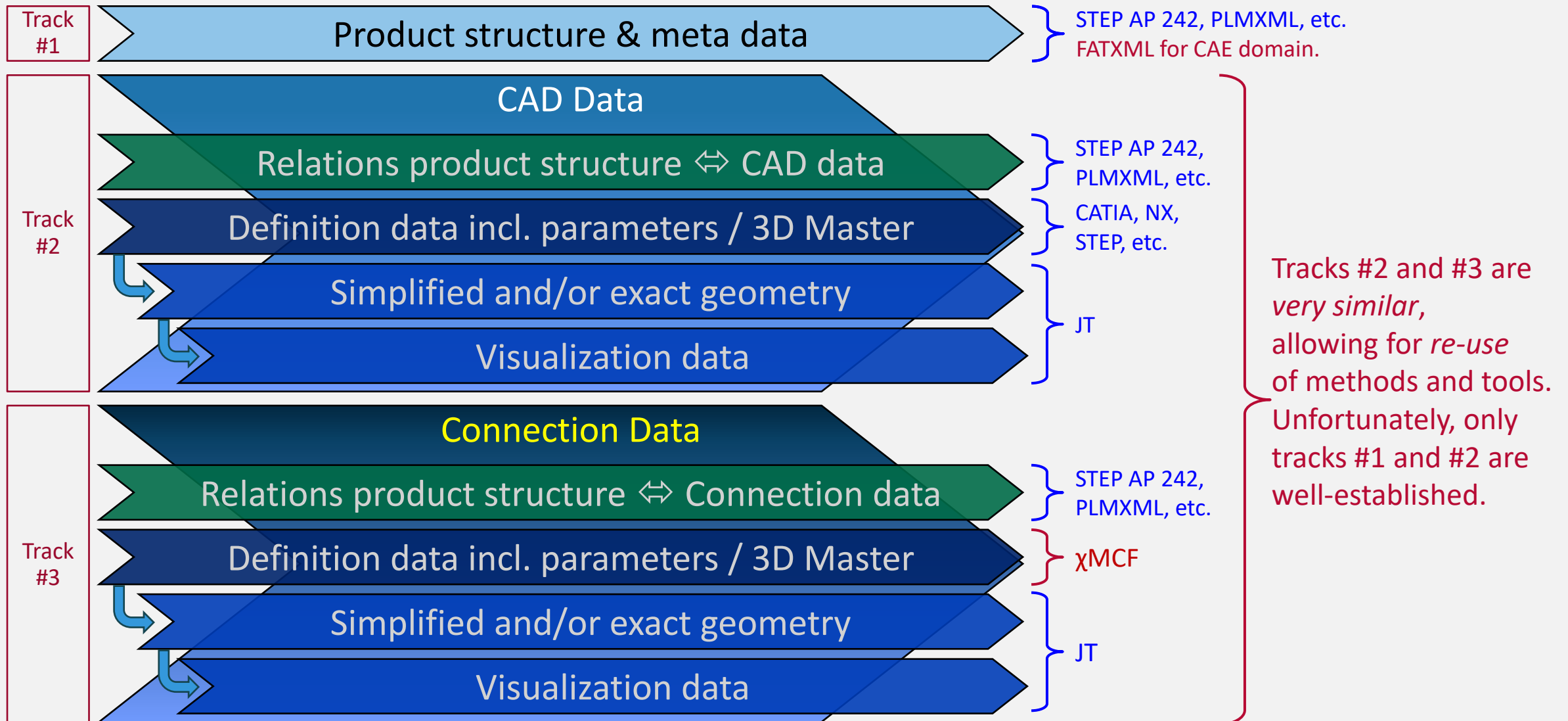
- FATXML erfüllt die gestellten Anforderungen.
- ➔ FATXML ist das Mittel der Wahl für zuverlässige und verlustlose Durchgängigkeit von Modell-Struktur und Meta-Daten im CAE-Prozess.

SYNTHESE & AUSBLICK

χ MCF & FATXML do their job

... aber wie integrieren sie sich
ins "Big Picture"?

Detailed *Solution* Architecture of PLM Data Tracks





Wie weit haben sich
 χ MCF & FATXML
inzwischen durchgesetzt?

Überblick Standard-Unterstützung

Software	FATXML	xMCF
<u>Pre/Post/Plug-Ins/integr. Systeme:</u>		
GNS – Animator 2.2.1	V1.2	---
BETA CAE – ANSA	V1.2 seit 16.0.0	V3.0 – in Test: Spot Welds, Robscans, Adhesives (0, 1, 2D), ... in Implementation: Bolts, Seam Welds in Planung: Rivets, Screws, Hemmings
Altair – HyperMesh	V1.2 in Implementation	V3.0 teilw. (Diskretisierung noch in Arbeit) Stand 13.09.16
T-Systems – MEDINA	V1.2 seit 9.0.2.0	V 1.1.0 seit 8.5
BETA CAE – µETA	V1.2 seit 16.0.0	Siehe ANSA ?
Siemens/NX CAD & Simcenter CAE	---	V3.0 in Diskussion
Siemens/Vistagy – Syncrofit	---	V2.1 teilw. / V3.0 in Diskussion
Siemens/LMS – Virtual.Lab	---	V2.1 teilw.
<u>Allgemeine Solver:</u>		
Dassault – Abaqus	Syntax: **XML – kombiniert mit Python	
LSTC – LS-DYNA	V1.1 – Syntax: *DATABASE_FATXML ID-Offsets & Include-Namen werden vom Solver ergänzt.	
MSC – NASTRAN	Syntax: METADATA FATXML.	
Altair – OptiStruct	Zugesagt – Geplante Syntax: METADATA FATXML.	
ESI – PAM-CRASH	Syntax: CDATA/NAME FATXML.	
INTES – PERMAS	Syntax: \$COMMENT	
Altair – RADIOSS	Zugesagt – Geplante Syntax: //PRIVAT/FATXML	
<u>Fatigue-Solver:</u>		
Magna Powertrain ECS – FEMFAT	---	V 1.1.0
HBM Prenscia – nCode Designlife	---	V 1.1.0 – Schweißlinien & Schweißpunkte nutzen PID-Gruppe.

*:= Ausnahme: LS-DYNA ergänzt Informationen.

Mögliche Synergie zwischen xMCF und FATXML

Software	FATXML	xMCF
<u>Pre/Post/Plug-Ins/integr. Systeme:</u>		
GNS – Animator 2.2.1	V1.2	---
BETA CAE – ANSA	V1.2 seit 16.0.0	V3.0 – in Test: Spot Welds, Robscans, Adhesives (0, 1, 2D), ... in Implementation: Bolts, Seam Welds in Planung: Rivets, Screws, Hemmings
Altair – HyperMesh	V1.2 in Implementation	V3.0 teilw. (Diskretisierung noch in Arbeit) Stand 13.09.16
T-Systems – MEDINA	V1.2 seit 9.0.2.0	V 1.1.0 seit 8.5
BETA CAE – µETA	V1.2 seit 16.0.0	Siehe ANSA ?
Siemens/NX CAD & Simcenter CAE	---	V3.0 in Diskussion
Siemens/Vistagy – Syncrofit	---	V2.1 teilw. / V3.0 in Diskussion
Siemens/LMS – Virtual.Lab	---	V2.1 teilw.
<u>Allgemeine Solver:</u>	<i>Solver berührt Inhalt nicht.* → Format-Version unerheblich.</i>	<i>Solver berührt Inhalt nicht. → Format-Version unerheblich.</i>
Dassault – Abaqus	Syntax: **XML – kombiniert mit Python	Syntax: **XML – kombiniert mit Python (undokumentiert)
LSTC – LS-DYNA	V1.1 – Syntax: *DATABASE_FATXML ID-Offsets & Include-Namen werden vom Solver ergänzt.	--- (Es gibt kein „DATABASE_xMCF“.)
MSC – NASTRAN	Syntax: METADATA FATXML.	Denkbare Syntax: METADATA xMCF (ungetestet)
Altair – OptiStruct	Zugesagt – Geplante Syntax: METADATA FATXML.	Denkbare Syntax: METADATA xMCF (ungetestet)
ESI – PAM-CRASH	Syntax: CDATA/NAME FATXML.	Denkbare Syntax: CDATA/NAME xMCF (ungetestet)
INTES – PERMAS	Syntax: \$COMMENT	Syntax: \$COMMENT (undokumentiert)
Altair – RADIOSS	Zugesagt – Geplante Syntax: //PRIVAT/FATXML	Denkbare Syntax: //PRIVAT/xMCF (ungetestet)
<u>Fatigue-Solver:</u>		<i>Solver interpretiert Inhalt. → Format-Version wichtig.</i>
Magna Powertrain ECS – FEMFAT	---	V 1.1.0
HBM Prenscia – nCode Designlife	---	V 1.1.0 – Schweißlinien & Schweißpunkte nutzen PID-Gruppe.

Wiederverwendung
mögliche

*:= Ausnahme: LS-DYNA ergänzt Informationen.