

„Ohne Software gibt es kein Wachstum mehr.“
Philipp Rösler, Bundesminister für Wirtschaft und Technologie, 21.5.2013

40. Scalable Software Business Models

Prof. Dr. Uwe Aßmann

Softwaretechnologie

Fakultät Informatik

Technische Universität Dresden

2019-0.4, 20-1-10

<http://st.inf.tu-dresden.de/teaching/saab>

- 1) Different Driving Factors
- 2) Service-based Business Models
- 3) Product-Based B2B Models

Obligatory

- ▶ [Cusumano] Michael A. Cusumano. Staying Power: Six Enduring Principles for Managing Strategy and Innovation in an Uncertain World. Clarendon Lectures in Management Studies. Oxford University Press, 2010.
 - Try to buy this book second hand, it is revolutionary.
 - Spend 20€ to win your lifetime's income!
- ▶ <http://www.drkarlpoppp.de/VeroeffentlichungenPublications.html>

References

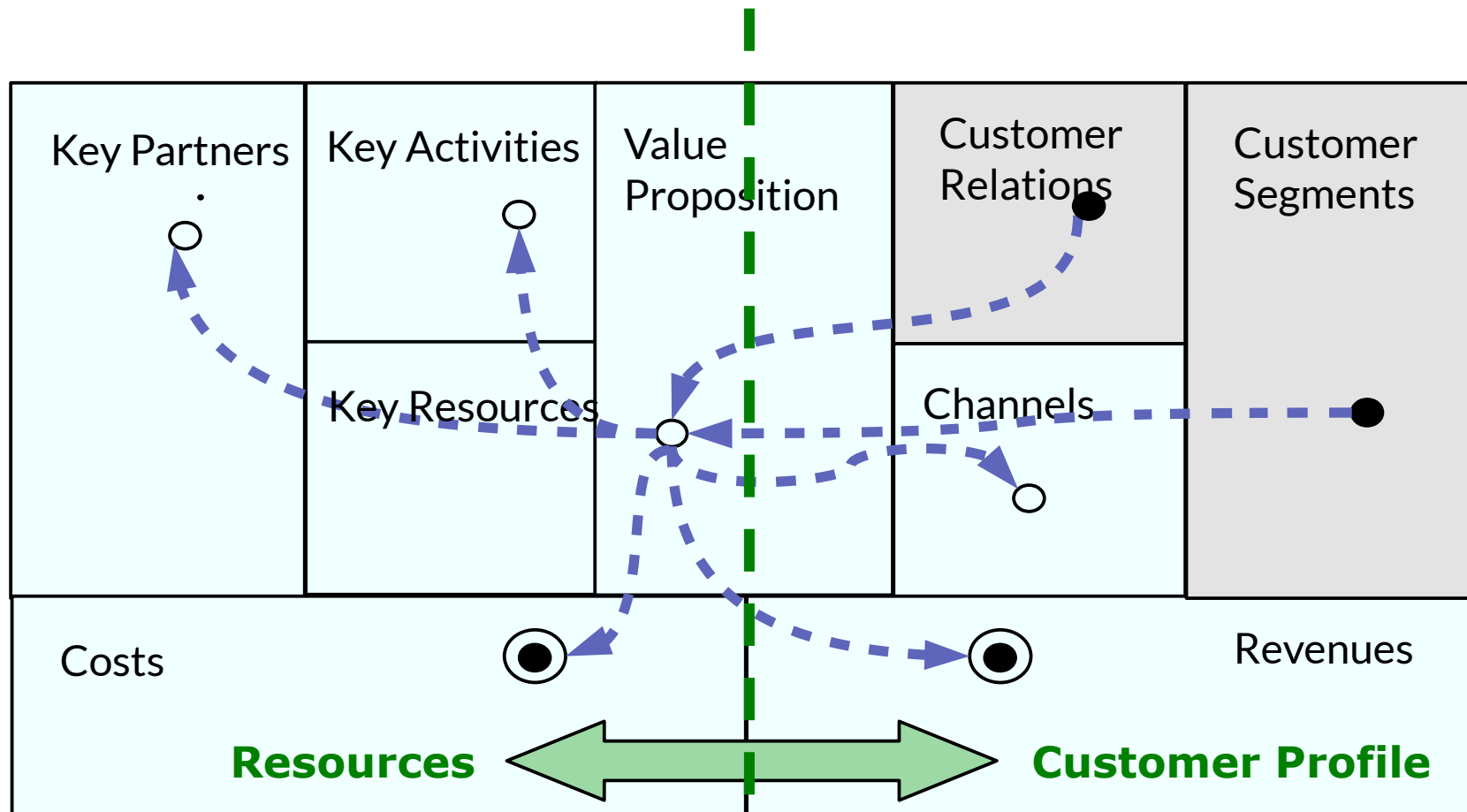
- ▶ Karl Popp. Software industry business models. IEEE Software, 28(4):26-30, 2010.
- ▶ [MassCustomization] Charles Krüger. Software Mass Customization. Biglever Software White Paper. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.6997&rep=rep1&type=pdf>
- ▶ [Scheer] August-Wilhelm Scheer. Unternehmen gründen ist nicht schwer. Springer. 2000. Honest book about ups and downs of ARIS.
- ▶ Klaus Schmid, Frank van der Linden. Software Product Lines in Action. Springer.

40.1 Different Factors Drive Business Models – How to Vary a Business Model Canvas

[BMG p 142]

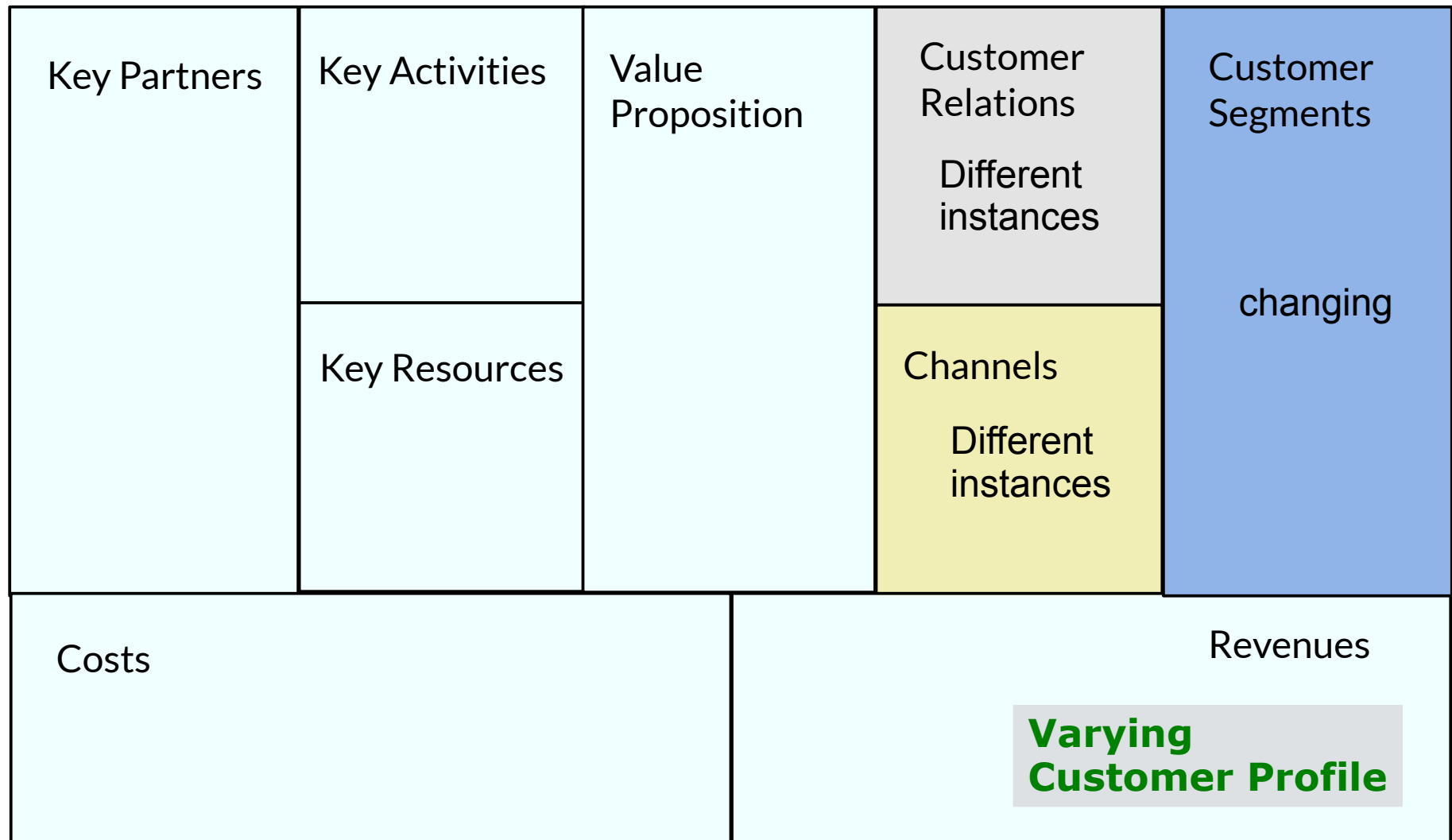
Customer-Driven Business Models

- ▶ From customers to value proposition and the rest
- ▶ Many Software BM use *product specialization* for disjoint groups of costumers
 - Setting up a customizable product or product line
 - Setting up a second product line in a product matrix



Customer-Varying Business Models for Services, Software Products and Product Lines

- ▶ Customers live in different domains, with different communities, habits, histories



Scale by

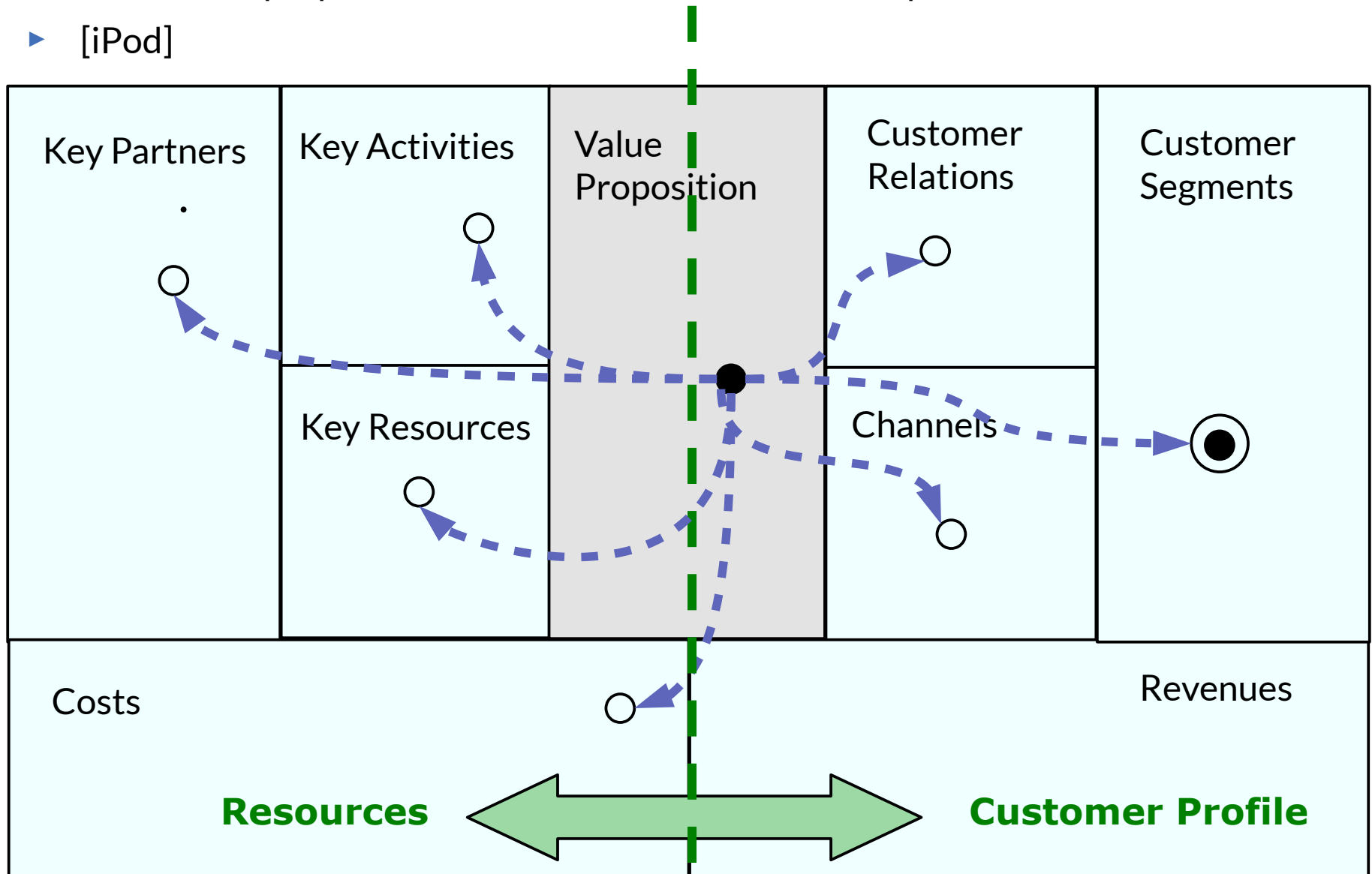
Variation of Customer Segments for Product Variation

- ▶ One of the most *scalable* software business models is **customer segmenting via domain-specific, regional, or customer-group specific product lines.**
- ▶ **Software reuse factors** should be high
 - To sell the reused components many times
 - → Component-based Software Engineering, Frameworking, Product Line Engineering

*Always attempt to derive a product variation
with a new customer segment
and a good software reuse.*

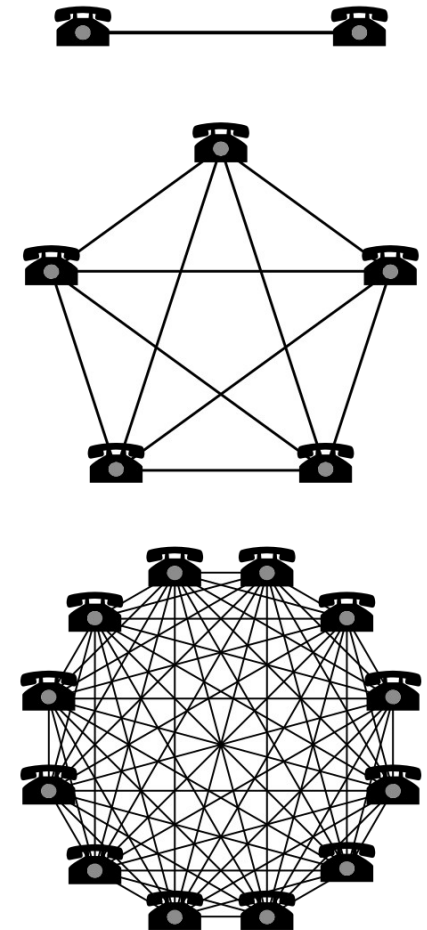
Value-Driven Business Models

- ▶ From value proposition to resource and customer map
- ▶ [iPod]



Scaling in Value-Driven Networks (Internet Marketplaces, Social Platforms)

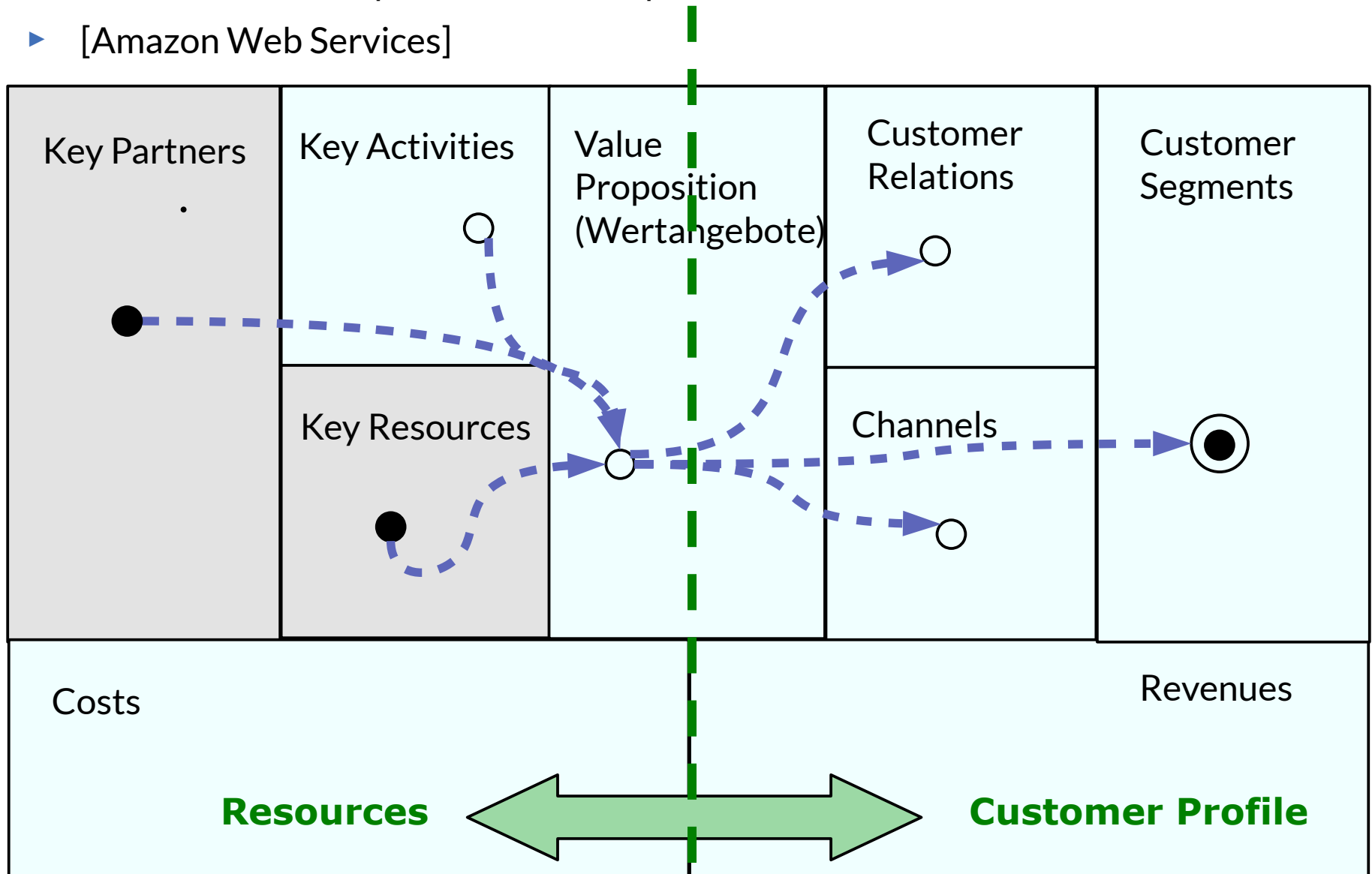
- ▶ https://en.wikipedia.org/wiki/Metcalfe%27s_law
- ▶ If the value proposition affects the *relation* of the customers, n^2 business opportunities result
- ▶ This is the business model of Facebook, Twitter, XING etc:
- ▶ Create a network first, sell later



Business models based on Value-Driven Networks are among the best-scaling business models

Resource-Driven Business Model

- ▶ From Resource map to customer map
- ▶ [Amazon Web Services]



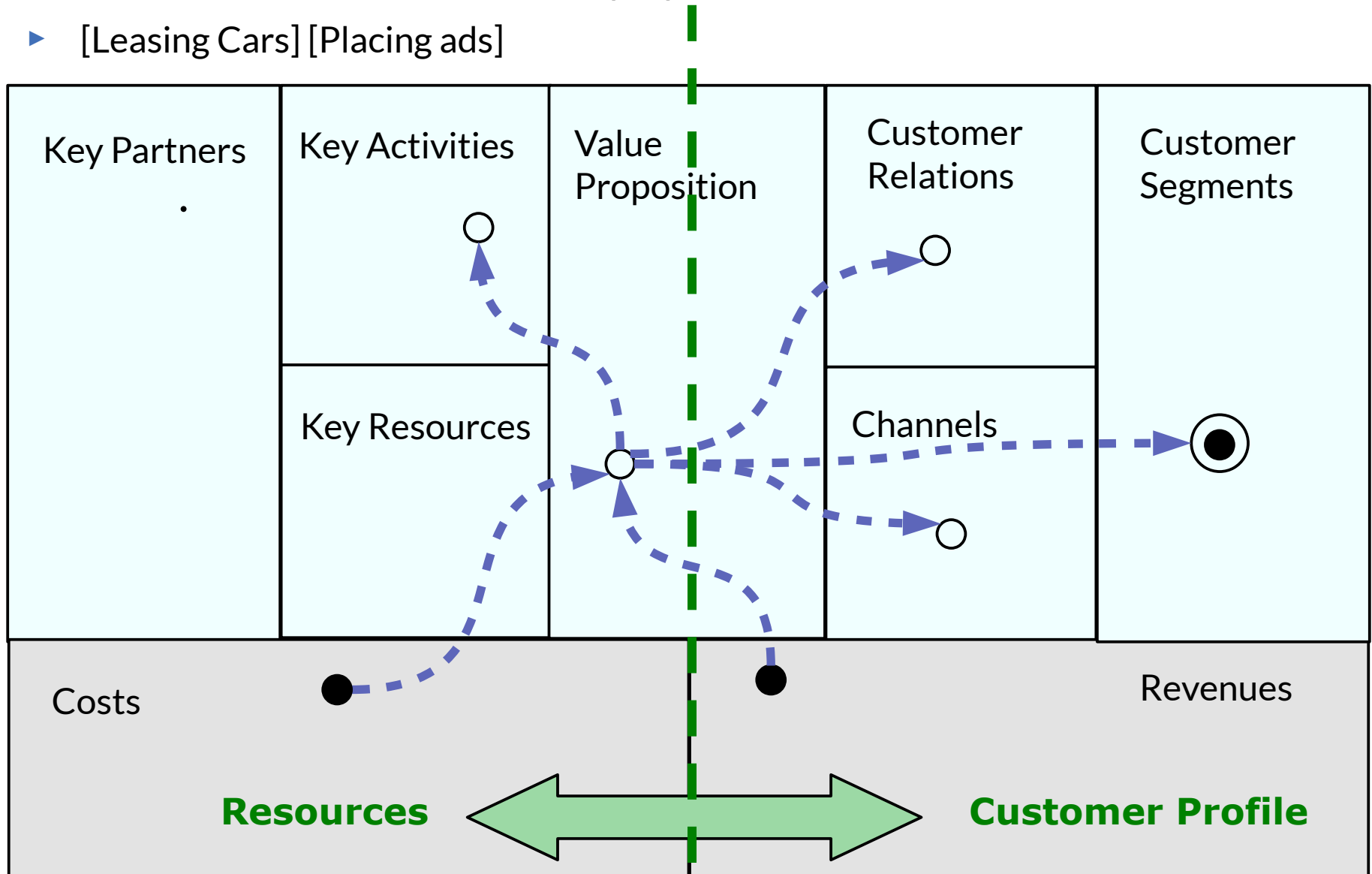
Technical Domains – Technology Variation-Driven Business Models

- ▶ The same user group can be reached by different technologies
- ▶ A **technical domain** is a technology which can be crossed with the **business domain** (customer segment).
- ▶ Product matrix models result (variation in 2 dimensions: business domain and technology)

Always attempt to derive another business model based on a technology variation crosscutting the customer segments.

Finance-Driven Business Models

- ▶ From costs and revenues to value proposition and customers
- ▶ [Leasing Cars] [Placing ads]



Finance Drives Variation

- ▶ A **finance-driven variation** sells a product with different pricing models (3rd dimension of variation):
 - Teaser for free, but with advertisement
 - Pay-per-use with paypal
 - Computer-based license
 - Flatrates
 - Location-based services with base product

Always attempt to derive a finance variation crosscutting the customer segments and the technology.

Finance: Instead of Sell Binaries

Closed-Source Software Business Models

- ▶ Leasing (where others buy)
- ▶ Rent (where others buy)
- ▶ Sell advertisements [Opera, Google]
- ▶ Sell directly, order via internet [Dell, Amazon]
- ▶ Sell later, hope to use the Metcalfe network effect [Twitter, Facebook]
- ▶ Sell via auction [ebay]

Open Source Software (OSS) Business Models

- ▶ http://en.wikipedia.org/wiki/Open_source
- ▶ **Free product** (“free taste”, “Versucherle”, “Köder”)
 - Give the product for free and **sell services, consulting, or apps**
 - Mould a market with the product
 - Ex. Adobe pdf with Acrobat Reader
- ▶ **Free framework**
 - Give the framework for free, create a community, and sell applications
 - Ex. IBM gives Eclipse for free, fosters a community, and many sell
- ▶ Release Politics
 - with union-fs (overlay); with browser; with portal
- ▶ **Micropayment**
 - Use micropayment companies for installation or run of a software (PayPal, ..)
 - Use Telecom billing
- ▶ Choose licences carefully
 - <http://creativecommons.org>
 - GPL is a virus that infects all extensions; LGPL not
 - FPI

Open Source Business Model “Free Taste” (dual-licensing, freemium)

- ▶ Free “taster” versions
 - Give out earlier version of the product for free
 - Sell the new or premium version (“freemium”)
- ▶ Examples
 - www.gentleware.com
 - NatSpec
- ▶ Free “community” versions
 - Give out a stripped version (e.g., only for 1 user, 1 database, ..)
 - Sell full version
- ▶ Free time-restricted versions
 - 1 month

Business Model “Versucherle”/“Köder”: Plugins under Dual Licensing

- ▶ Premium: [BMG] p. 108
- ▶ Companies can make plugins for OSS tools under dual licensing
 - Thunderbird, Firefox, OpenOffice, Eclipse, ...
- ▶ Example: Quicktext Thunderbird extension <http://extensions.hesslow.se/>
 - QuickText is free
 - QuickText Pro is commercial
- ▶ Advantage: Platform has already many users and a large market

Deriving Secondary from Primary Business Models

- ▶ From a primary value-driven or customer-driven business model, secondary ones can be derived
 - **Customer-driven models** vary the customer segment
 - **Resource-driven models** can be derived to use the resources more efficiently
 - **Finance-driven models** can be derived to re-use a good value proposition with other pricing and costs
 - [Hilty drilling machines]

Always attempt to derive a secondary BM from your primary one by varying along one of the variation dimensions.

Basic Forms of Business Models

- ▶ Product (with maintenance contract)
- ▶ Product with parameterization and mass configuration
- ▶ Product with piggy-pack Service

- ▶ Product Line (variation in 1 dimension)
- ▶ Product Matrix (variation in 2 dimensions)
- ▶ Product Cube (variation in 3 dimensions)
- ▶ Product Platform with Apps/Complements/Plugins and Services

- ▶ Individual software solution (with maintenance contract)

- ▶ Service
- ▶ Service Family
- ▶ Service based on Value Driven Network
- ▶ Service Matrix (variation in 2 dimensions)
- ▶ Service Cube (variation in 3 dimensions)



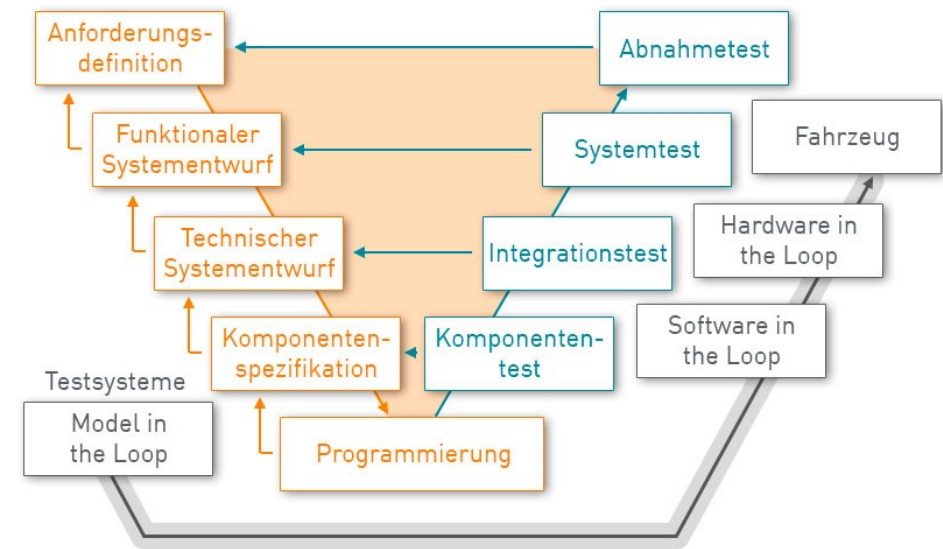
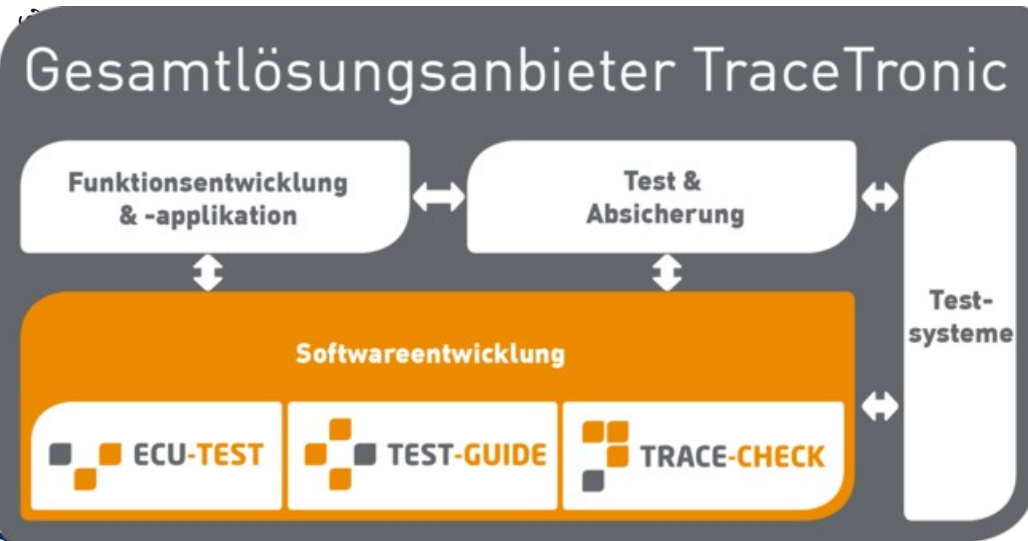


40.2 Service-Based Business Models

40.2.1. Services (“Leistungen”, “Solutions”) can be based on Products

- ▶ <http://www.tracetronic.de/leistungen/>
- ▶ TraceTronic (Gittersee) has several products for motor tests
- ▶ On top, they offer test projects, construction and integration into further test systems, function development for motors
- ▶ Testing uses the automotive V-model: lots of spots for service projects
 - Requirements engineering (elicitation, checking...)
 - Specification
 - Design and Testing
 - Seminars (Schulung)

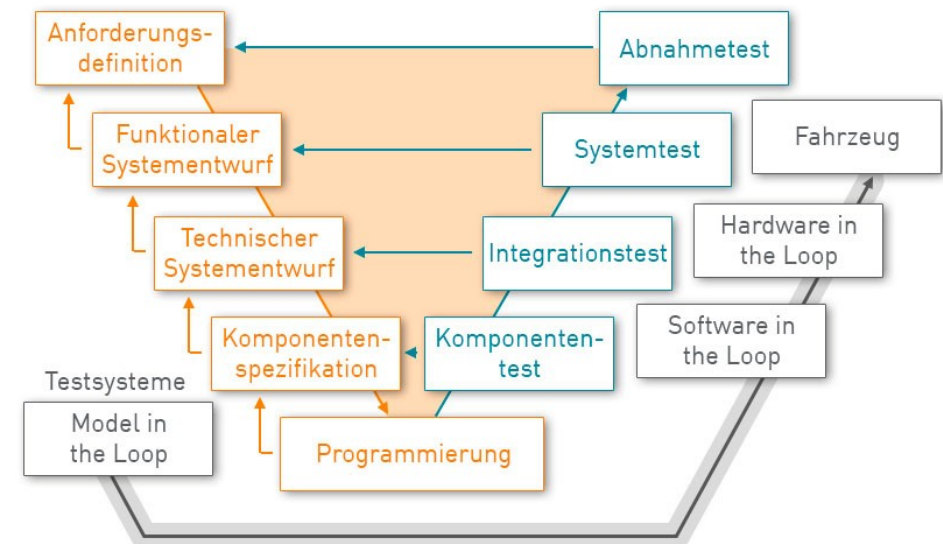
© Prof. Uwe Aßmann



Business Model “Service Family along V-Model” Salt Solutions

23 Software as a Business

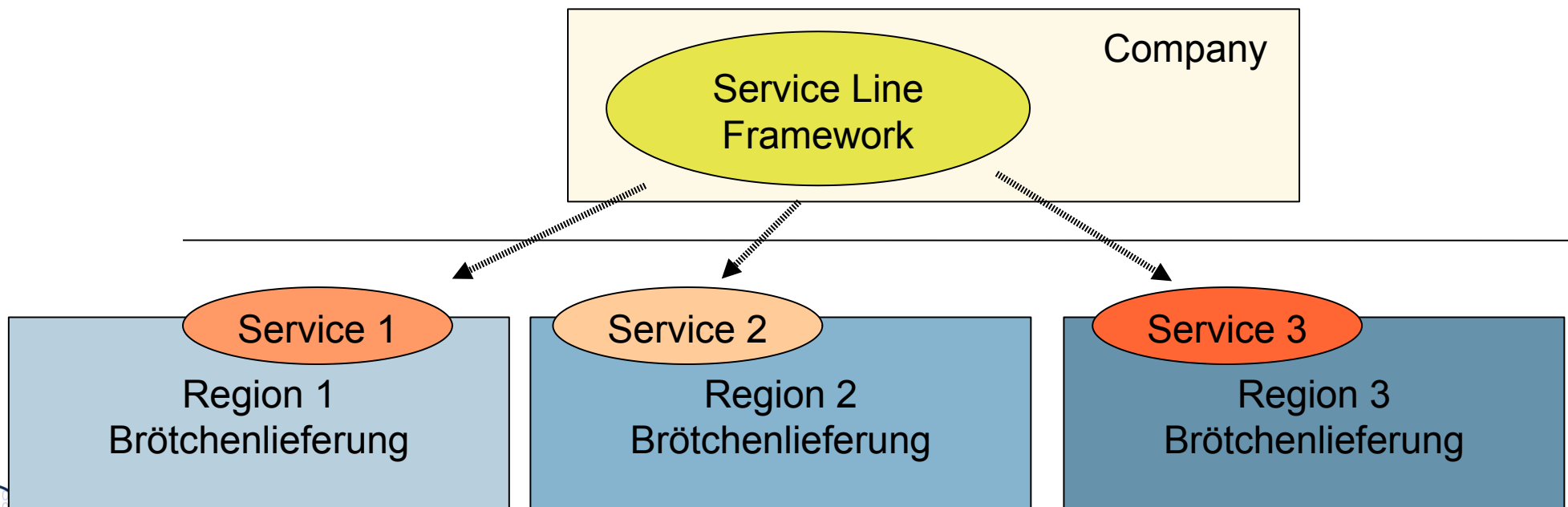
- ▶ For any software company, services can be structured along the typical tasks of the V-model
- ▶ <https://www.salt-solutions.de/leistungen.html>
 - Quality management
 - Requirements engineering
 - Testing
 - Certification
 - Consulting
 - Conception (Design, Processes, Specification)
 - Implementation
 - Support



- ▶ <http://www.tracetronic.de/leistungen/>

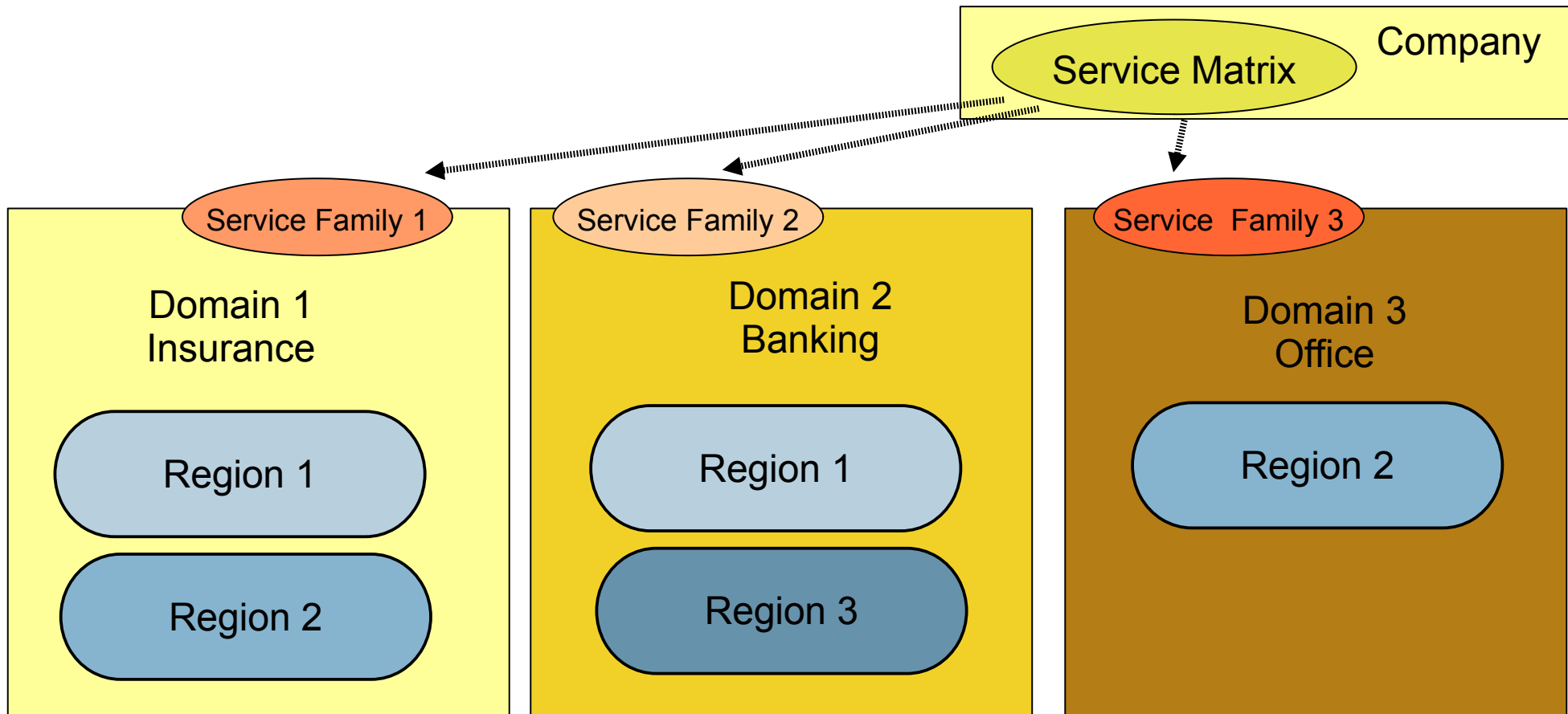
40.2.2 Business Model “Regional Service Family”

- ▶ One dimension to vary can be the region of sales
 - Develop a service in a region (regional business model, regional customer segments)
 - Replicate the service to other regions (regional porting)
- ▶ Example: www.Morgengold.de, www.it-sax.de (pludoni.de), www.lieferando.de
- ▶ Franchising can be used to spark small enterprises
- ▶ The same software can be used in every region, if customized appropriately



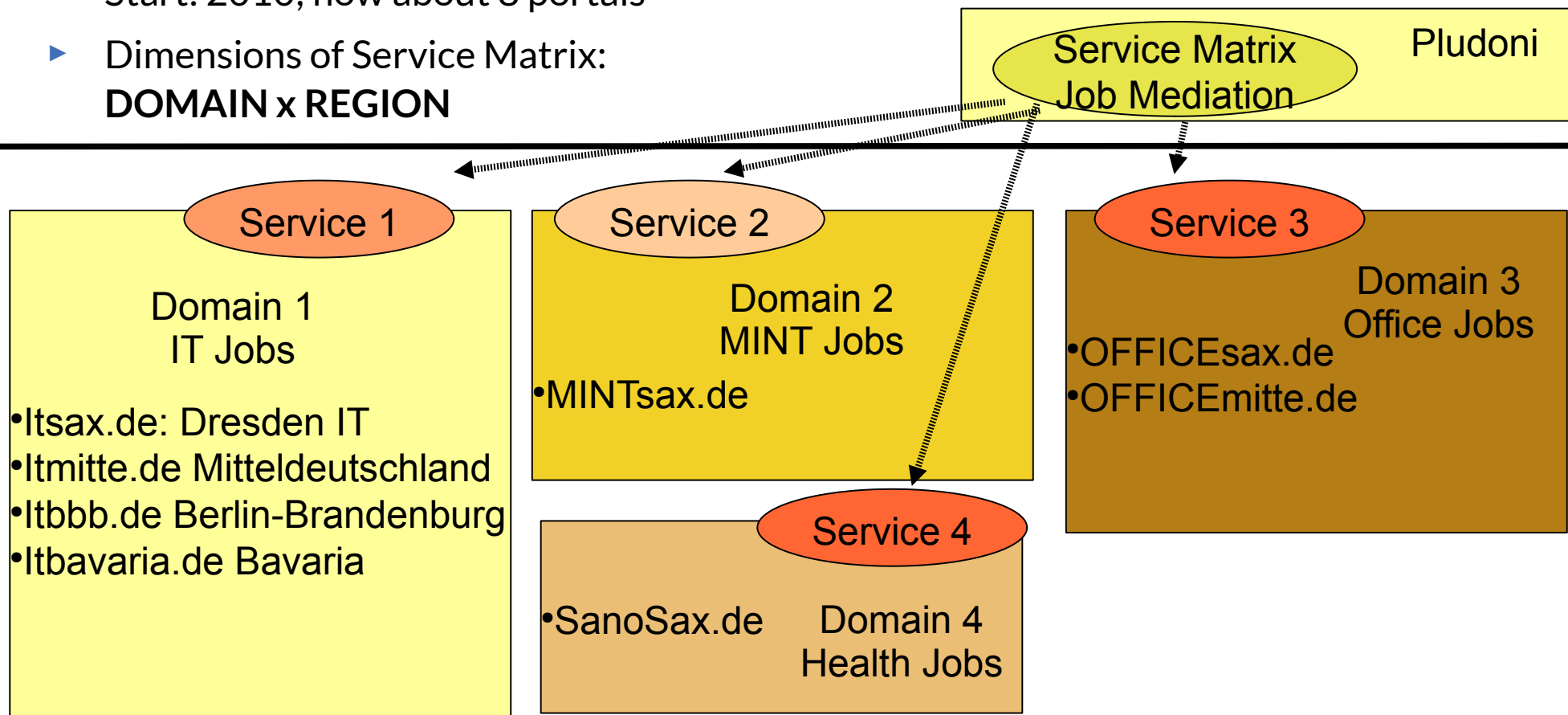
40.2.3. Business Model “Domain-Region Service Matrix”

- ▶ Develop a service in a two or three dimensions
 - Region
 - Business Domain
 - Technical Domain



Pludoni's Business Model “Domain-Region Service Matrix” with “Smart Companies”

- ▶ www.it-sax.de (www.pludoni.de): a portal for mediating (brokering) applications and job offers in a region (regional business model)
 - Trusted web community (“club of companies”) with recommendations
- ▶ Founder Jörg Klukas ported it to many other regions and professional domains
- ▶ Start: 2010; now about 8 portals
- ▶ Dimensions of Service Matrix:
DOMAIN x REGION

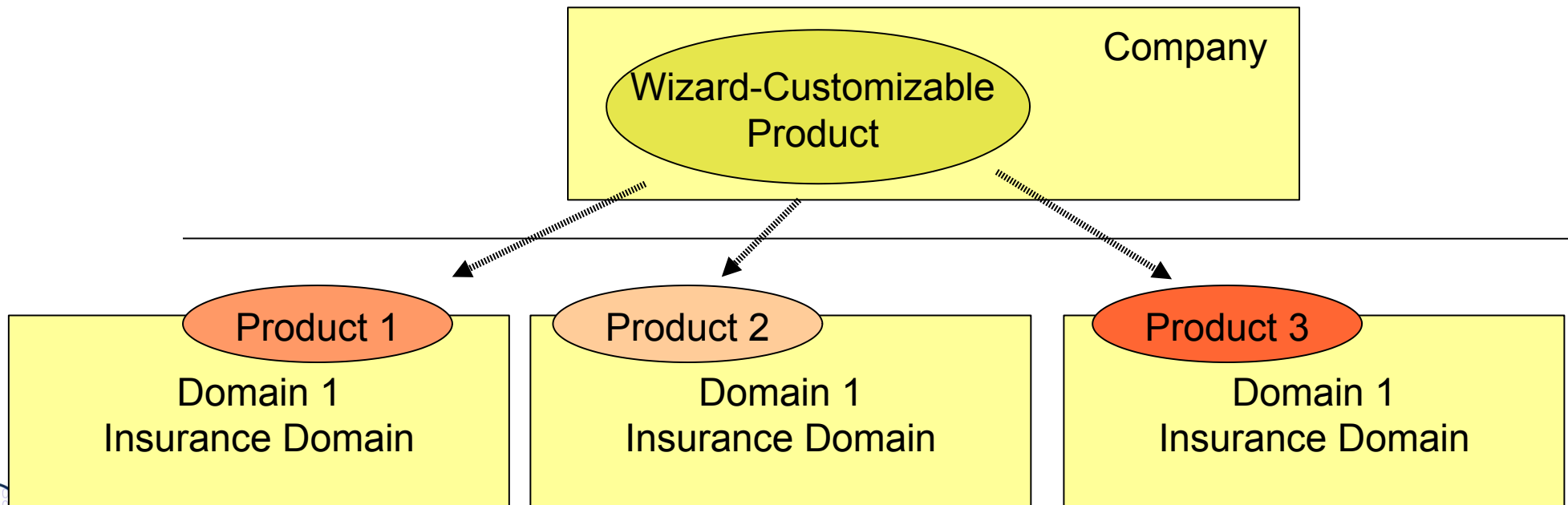




40.3 Product-Based B2B Business Models

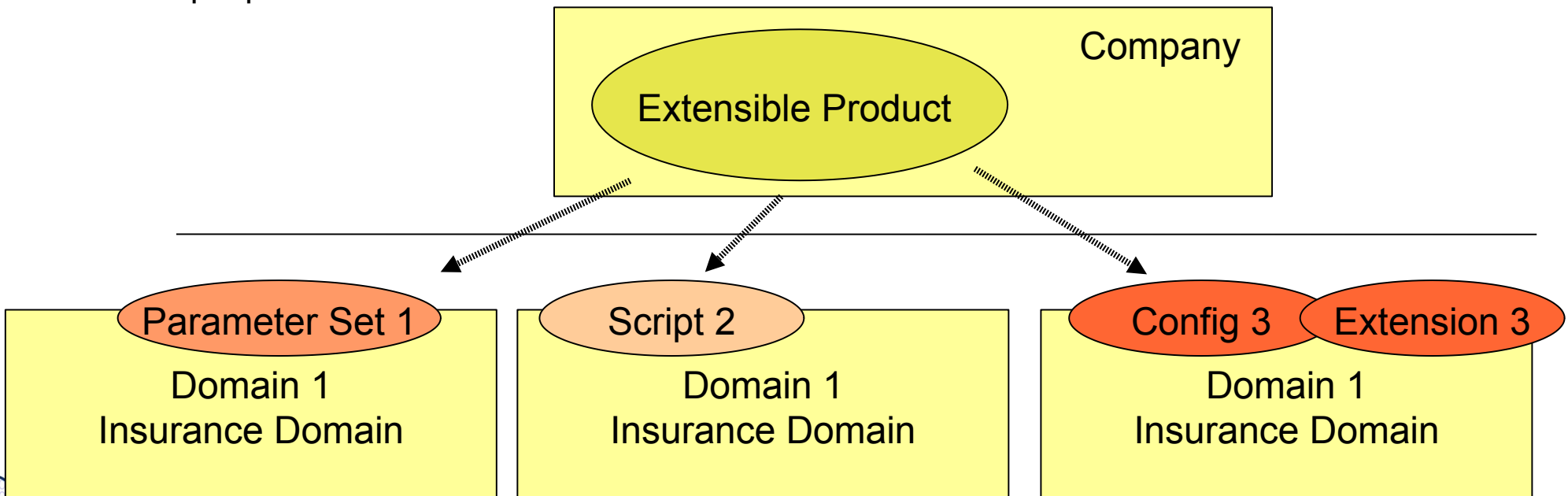
Business Model “Mass-Customizable Product”

- ▶ Make the product **customizable (by end user)** [MassCustomization]
 - Customize by interactive Wizard simplifying the customization
 - Keep the product customization technique as company secret
- ▶ Intense interaction with customers required to ensure that the mass customization is simple enough



Business Model “Parameterizable Software Product”

- ▶ Design an extensible product in-house, **parameterizable by experts**
 - Parameterize by scripts in a scripting language, domain-specific language, or complex form mechanism (XML)
 - **Extend** by Plugin (Extension, Complement); Requires a *component model*
- ▶ Extension by product owner or third party (partner or supplier)
- ▶ Know how: instantiate new products with *different extended functions (features)*
 - Keep the product extension technique as company secret
- ▶ This is the standard way for product-oriented companies, how to scale their business to more than 15 people



Tracetronic ECU-Test, a Parameterizable Domain-Specific Product

- ▶ Gittersee, Dresden, Germany
- ▶ <http://www.tracetronic.de/produkte/ecu-test/>
- ▶ Motor testing software
 - Writing and composing test scripts for motor runs in simulators or cars
 - Test script libraries
- ▶ Model-in-the-loop (MIL), Hardware-in-the-loop (HIL), Software-in-the-loop (SIL)
- ▶ Adapters for other test tools
- ▶ Growth: 1->100 employees in 10 years

**Domain-specific products and product lines fill a domain-specific need
(a need of a domain-specific customer segment)!**

Tracetronec ECU-Test

The screenshot displays the ECU-TEST 6.0 software interface. The main window is titled 'Editor' and shows a test case for 'CrankSensor'. The test case is structured as follows:

#	Action / Name	Parameter	Expectation / Value
1	Initialization		
2	MODEL-Write: CTRL_ABTRIEB/KL15 [0][1]/Value	PHYS(don't care)	1
3	WAIT	500 ms	
4	DIAG(EDIABAS): Engine/FS_LOESCHEN		JOB_STATUS: E('OKAY')
5	DIAG(EDIABAS): Dashboard/FS_LOESCHEN		JOB_STATUS: E('OKAY')
6	Engine start and driver request		
7	EngineStart	NMotMin=NMotMin	
8	Comment		'Driver request'
9	MODEL-Write: CTRL_ABTRIEB/Fahrerwunsch [%]/Value	PHYS(don't care)	DriverRequest
10	WAIT	3000 ms	
11	Comparison normal state		
12	NMot_Read		-> NMotDeviation
13	Calculation	NMotDeviation	<= 150
14	Fault crank sensor		
15	Electric failure simulation: FIU-1	E_P_KWG	BROKEN
16	WAIT	500 ms	
17	Comment		'Comparison fault state'

The interface also includes a left sidebar with 'Actions' and 'Configurations' panels, and a bottom section for 'Variables' and 'Mapping'.

	Variables	Mapping	Name	Initial value	Current value	Description
0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	DriverRequest	10 <Numeric>		Acceleration pedal in %
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	FortCrank	10900 <Numeric>		Error place number for c...
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	FortExpected	[10100, 10250, 1030...		allowed (expected) faults
3	<input type="checkbox"/>	<input type="checkbox"/>	FortList	[] <PyObject>		Created in EDIABAS tes...



Tracetronec ECU-Test

The screenshot shows the TRF-Viewer application window. The title bar reads 'TRF-Viewer - C:\Data\ECU-TEST6\TestReports\CrankSensor_2014-11-20_123024\report.trf'. A green status bar at the top indicates 'Authenticity check of the test report performed successfully. Test report was not manipulated.' The main area displays a test report table with columns for step number, time, action, name, and value. A dialog box titled 'Expansion filter' is open, showing a list of filters to be applied to the report.

#	Time [s]	Action	Name	Value
1	0.000	Initialization		
2	0.001	MODEL-Write	CTRL_ABTRIEB/KL15 [0]1]Value	1
3	0.005	UTILITY	Wait	500 ms
4	0.510	DIAG(EDIABAS)	Engine/FS_LOESCHEN	JOB_STATUS=OKAY
5	0.516	DIAG(EDIABAS)	Dashboard/FS_LOESCHEN	JOB_STATUS=OKAY
6	0.523	Engine start and driver request		
7	0.523	PACKAGE	EngineStart	C:\Data\ECU-TEST6\Packages\Lib\EngineStart.pkg
7.1	0.526	MODEL-Write		
7.2	0.530	UTILITY		
7.3	1.034	MODEL-Write		
7.4	1.038	MODEL-Read		
7.5	1.049	MODEL-Write		
8	1.053	UTILITY		
9	1.054	MODEL-Write		
10	1.057	UTILITY		
11	4.067	Comparison		
12	4.068	PACKAGE		
12.1	4.072	MODEL-Read		
12.2	4.081	MEAS-Read	Engine/n	710
12.3	4.090	DIAG(EDIABAS)	Engine/STATUS_MESSWERTE	2 result sets, 3 measured values
12.4	4.094	UTILITY	Calculation	NMotDiag[0] -> 710
12.5	4.096	BUS-Read	A-CAN/ENG_RPM	709
12.6	4.114	UTILITY	Calculation	max(NMotModel, NMotMeas, NMotDiag, NMotCAN) - mi
13	4.119	UTILITY	Calculation	NMotDeviation -> 111
14	4.122	Fault crank sensor		

Expansion filter

Selection of filters that classify which steps shall be expanded (OR linked)

- Evaluation == FAILED
- Evaluation == ERROR
- Execution depth <= 3

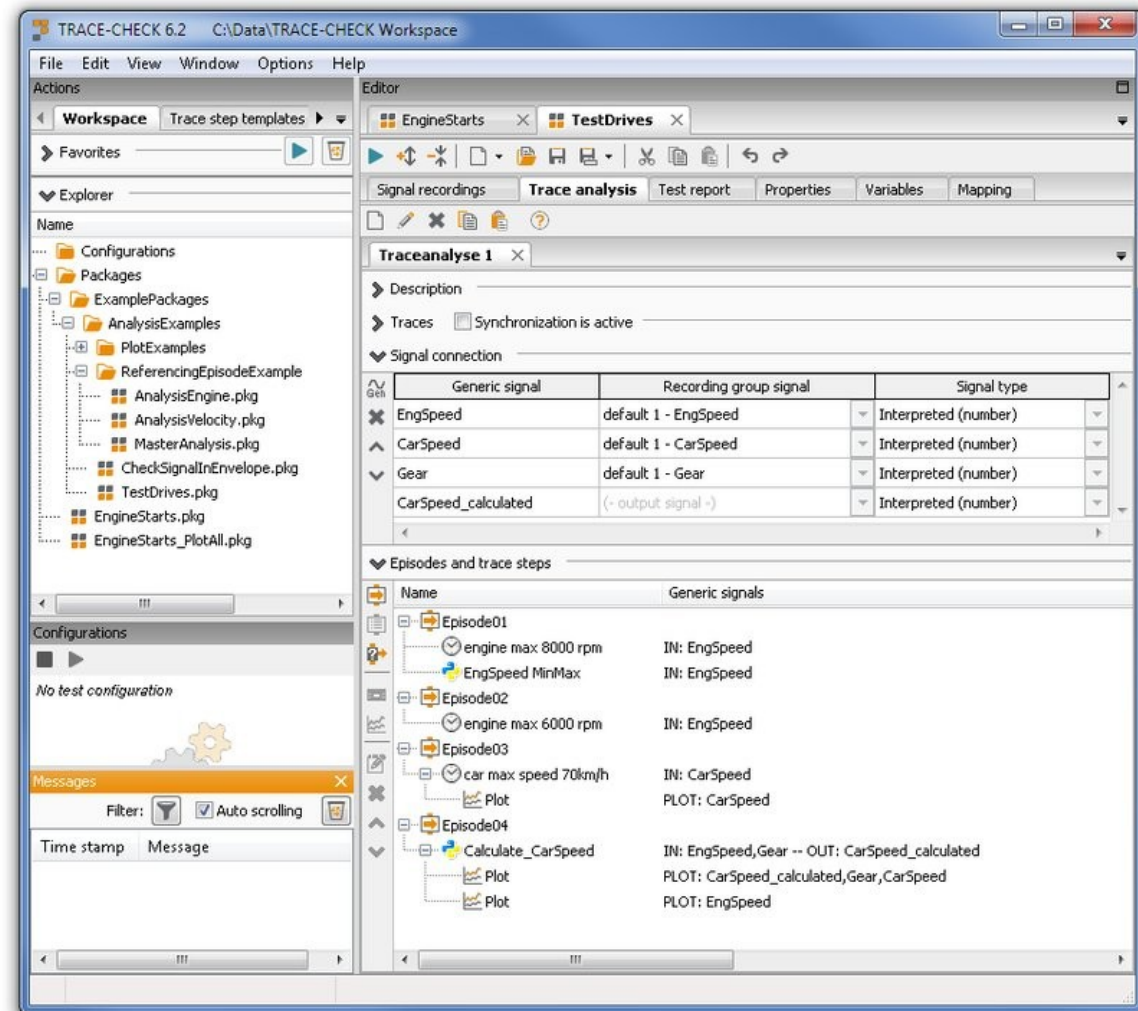
Buttons: Settings, Ok, Cancel



Tracetronic 2nd Product: Trace-Check

33 Software as a Business

- ▶ Monitoring and evaluating test traces with temporal logics (path expressions over time, see model-checking courses)
- ▶ Modular trace analysis
- ▶ Reporting
- ▶ Works with EXAM of VW
- ▶ <https://www.exam-ta.de/>



Tracetronic Customer Channels

- ▶ http://www.tracetronic.de/cms/data/docs/pdf/Datenblatt_ECU-TEST.pdf
- ▶ Tracetronic is a domain-specific company (automotive domain)
 - supplier to big car OEM (such as BMW)
 - It collaborates with other suppliers, such as ETAS or Vector
 - Many adaptors to other tools

***Domain-specific products and product lines fill a domain-specific need
(a need of a domain-specific customer segment)!***

Carl Zeiss Innovationszentrum für Messtechnik

Product: ZEISS Involute

- ▶ Dresden product for data analysis of gearwheels, e.g. wind power plant gearwheels
- ▶ First ZEISS software product, more than 100k customers worldwide
- ▶ <http://www.zeiss.de/izm/involute.html>
- ▶ Every year, ZEISS Dresden sponsors the Diplompreis Informatik for the department:
- ▶ <http://www.zeiss.de/izm/diplompPreis.html>

The screenshot displays the 'Nominal data and Tolerances' dialog box in the ZEISS Involute software. The interface is divided into two main sections: 'Gear data' and 'Tolerance of tooth thickness'. The 'Gear data' section includes a 'check tolerances' checkbox and a 'K-chart' plot showing gear tooth profiles with various tolerance zones. The 'Tolerance of tooth thickness' section contains a grid of input fields for various gear parameters and their tolerances.

Parameter	Value
teeth: z	14.00000
teeth in sector: z _S	0
module: m _n	20.0000000
add. mod. factor: x	0.50000
angle: α _n	20.00000
helix angle: β	0.00000
parameter: d _a	338.00000
flank direction:	spur
parameter: d _f	263.11403
face width: b	150.00000
profile: z ₀	0.00000
type of gear:	external
diam.: d _b	263.1139
pitch diam.: d	280.0000
unit:	mm
quality:	DIN_3962

Counting of Teeth:
average length profile: 0.00000
average length lead: 0.00000
positiv
teeth

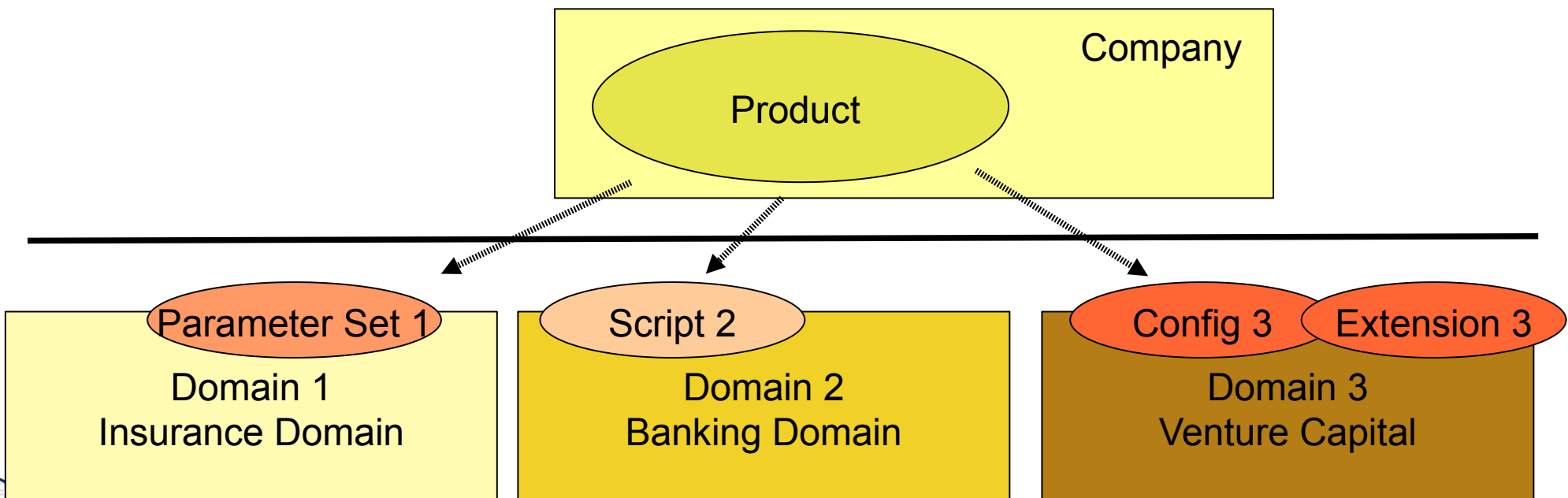
Buttons: Load..., Save, Save as..., Ok, Cancel, Help

Business Model

“Parameterizable Domain-Portable Technical Product”

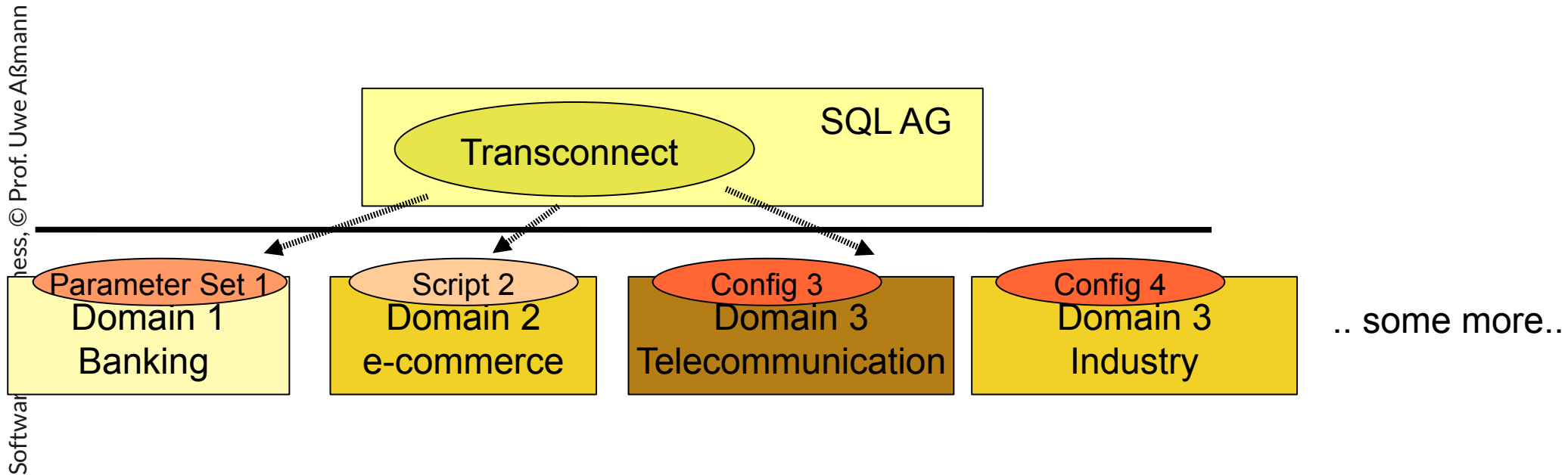
36 Software as a Business

- ▶ Produce your product for several domains
- ▶ Make it **parameterizable, customizable, extensible** (by experts of different domains)
 - Offer a domain-unspecific scripting language
 - *A technical component model*
- ▶ Problems: hard to sell, because customers appear in different domains (no closed customer group)
 - CRM is very important to memorize the details of the customer



SQL AG Transconnect Integration Suite

- ▶ <http://www.sql-ag.de/transconnect.html> Dresden, Franklin-Strasse
- ▶ Non-domain-specific, technical product: can be applied in many domains
 - Does not imply a domain-specific customer binding
- ▶ Data connector (integrator) between systems; many adapters



The End

- ▶ Explain the business model “domain-region service matrix”
- ▶ Explain the steps how to arrive from a product to an extensible product
- ▶ Explain how the BMC and the LeanCanvas help to find customer segments for a product line
- ▶ Compare the concepts of a Product Line, Product Matrix, and a Product Cube
- ▶ Explain how the products of a product line can be assessed with the BMC assessment process