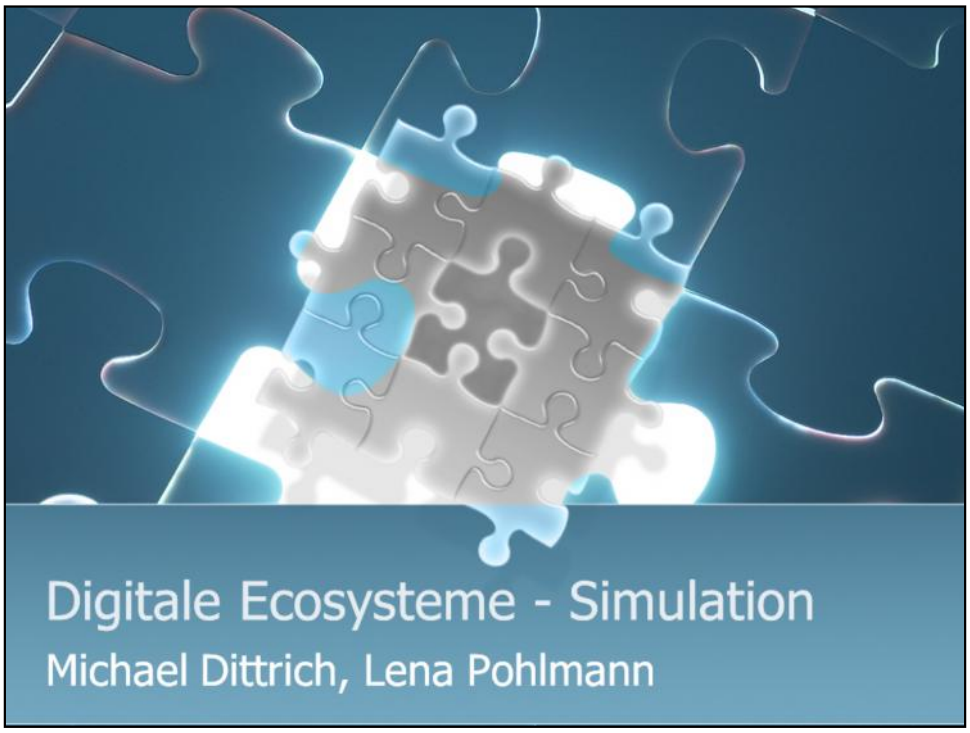
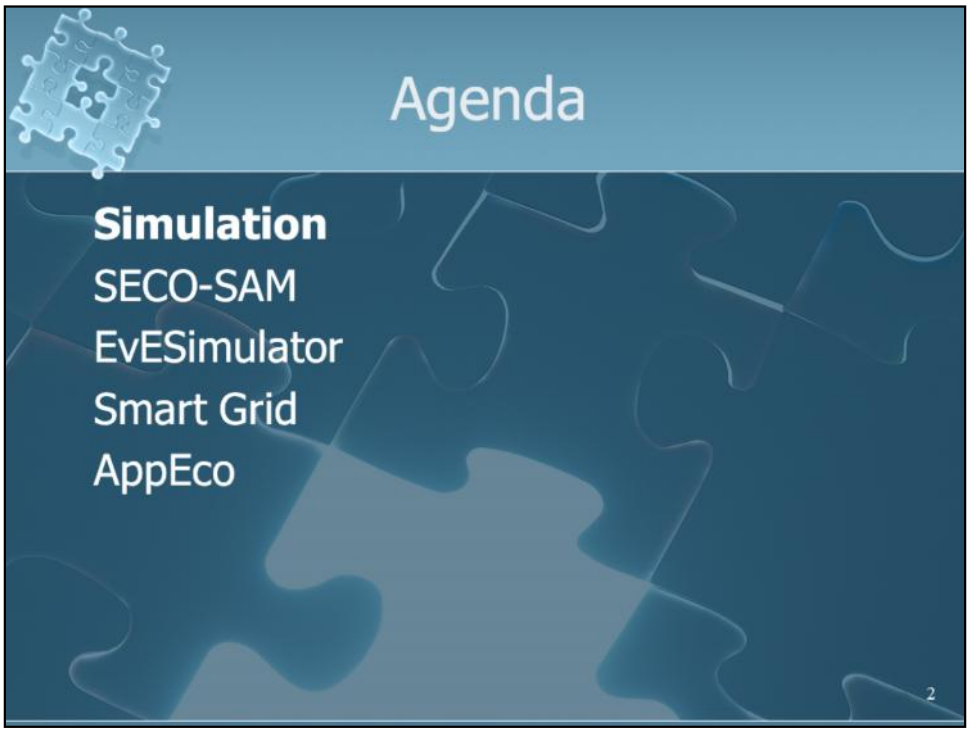
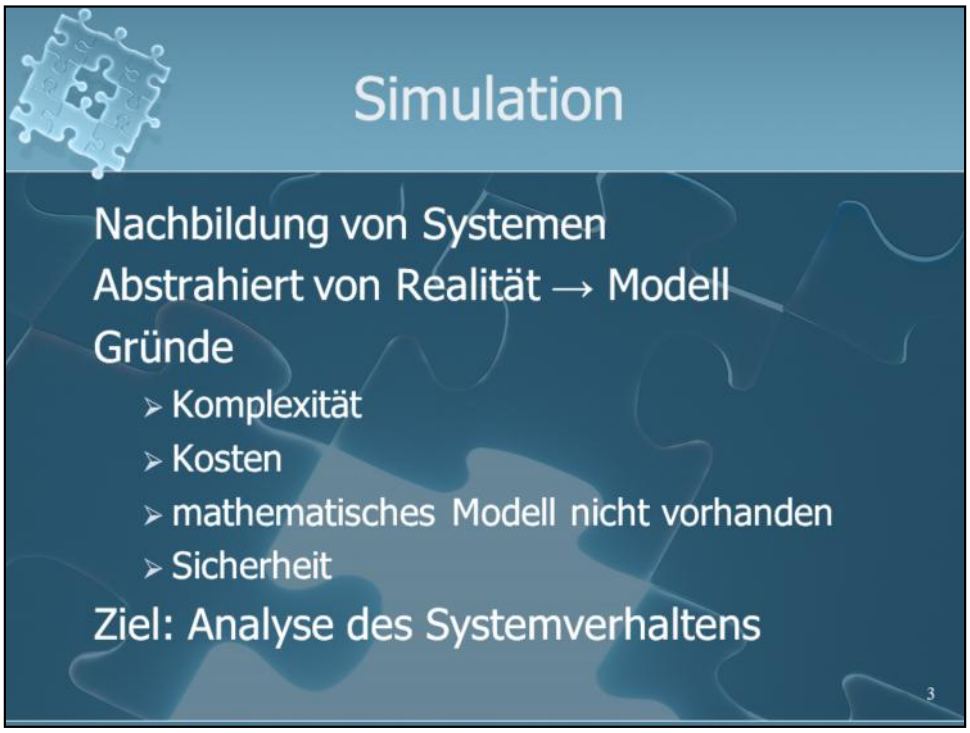


Digitale Ecosysteme - Simulation

Michael Dittrich, Lena Pohlmann


Agenda
Simulation
SECO-SAM
EvESimulator
Smart Grid
AppEco
2



Ein Simulation ist die Nachbildung eines Systems basierend auf einem Modell, welches vom realen zu simulierenden System abstrahiert.

Gründe für Simulationen sind: hohe Komplexität, hohe Kosten, Sicherheit einer Simulation oder dass es kein mathematisches Modell für ein bestimmtes System gibt.

Achtung: Simulationen sind eher „Vermutungen“.

Ziel einer Simulation ist die Analyse des Systemverhaltens unter bestimmten Bedingungen.

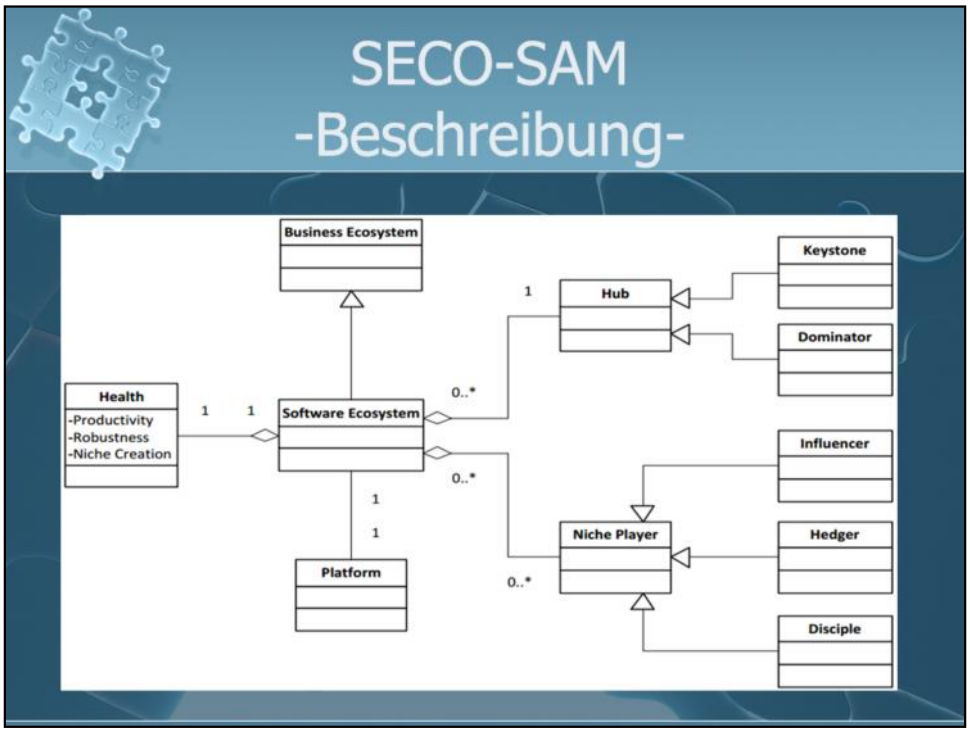

Agenda
Simulation
SECO-SAM
Definition
Beschreibung
Charakteristiken
Ergebnis
EvESimulator
Smart Grid
AppEco
4

Erste SECOs:
Desktop Betriebssysteme in den 80er.
Plug-In Software der 90er.
Aktuelles Beispiel der Apple Appstore.

Kunden können entweder die teilnehmenden Unternehmen selber sein oder Endkunden (Appstore)

SECO-SAM = Software Ecosystem Strategy Assessment Model

Ein Model als Beispiel was man alles an einem Software Ecosystem erfassen und quantifizieren kann.



Hub ist der Besitzer/Anbieter eines Software Ecosystem. (Bsp.: Microsoft ist hub des Windows Ecosystem)

Während ein ‘keystone’ versucht Mehrwert für sich und den Rest des ecosystem zu erzeugen, ist das Ziel eines ‘Dominator’ möglichst viel Wert für sich aus dem Ecosystem zu ziehen, in Folge dessen er es nach und nach zerstört.

Niche Player machen den Rest des Ecosystems aus.

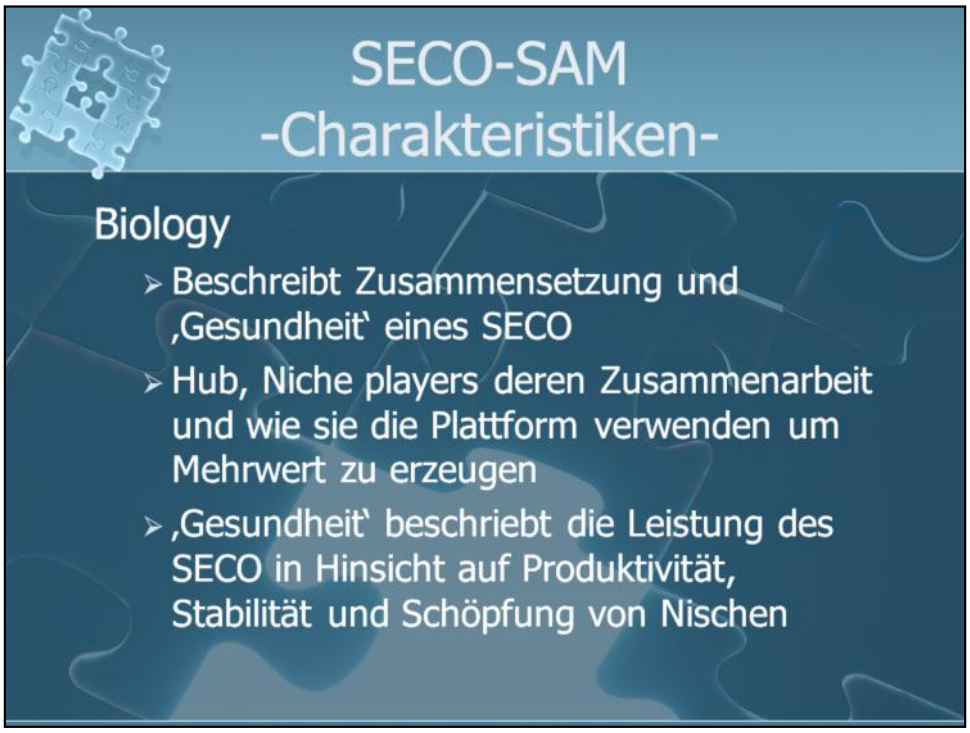
-Influencer : Nimmt früh am Ecosystem teil und beschränkt sich auf eine shape (keystone) strategy

-Hedger: Entwickelt Produkte für shaping (keystone) platforms

-Disciple: Entwickelt exklusiv für ein shaping (keystone) platform

Niche player haben allein wenig Einfluss auf das Ecosystem. Machen zusammen aber den größten Teil an 'Masse' und 'Varianz' aus.


SECO-SAM
-Charakteristiken-
Charakteristiken
Biology
Lifestyle
Environment
Health Care Organizations


SECO-SAM
-Charakteristiken-
Biology
Beschreibt Zusammensetzung und ‚Gesundheit' eines SECO
Hub, Niche players deren Zusammenarbeit und wie sie die Plattform verwenden um Mehrwert zu erzeugen
‚Gesundheit' beschriebt die Leistung des SECO in Hinsicht auf Produktivität, Stabilität und Schöpfung von Nischen

Strategic Level

-SECO Vison: Soll niche player zeigen in welche Richtung sich das SECO entwickeln soll. Ermutigt niche players dazu am SECO teilzunehmen um so dem SECO zu größerem Erfolg zu verhelfen.

-Plattform Strategy: Aus der SECO Vision erzeugter Plan, meist für die nächsten 5 Jahre.

-Stability: Es ist wichtig das Vision und Strategie stabil bleiben. Ansonsten werden niche players verunsichtert. Was dazu führen könnte, dass sie zu einem anderen SECO wechseln.

-Reputation

Tactical Level

-Auferlegte Barrieren: Können helfen nur ausgewählte Unternehmen am SECO teilzunehmen. (Beispiel Apple Appstore mit Application Approvel). Zu niedrige Hürden können der Stabilität gefährden. Zu hohe Hürden verringern die Innovation.

-Orchestration techniques: Grenzen/Regeln/Zertifikate können Entwicklern helfen zu erkennen was von ihnen erwartet wird und so die Produktivität steigern.

-Platform planning

-Community Building: Ermöglichen/Ermutigen der niche player miteinander zu kommunizieren.

-Sharing

-Openness: Open Standards/Formats/Source können den niche players helfen miteinander zu arbeiten.

Operational Level

-Research&Development: Investitionen um neue Innovationen zu erzeugen.

-Marketing & Sales

-Support & Service

Beispiele

Suppliers: Ein Anbieter eines Services/Software die essential für das SECO ist zieht sich zurück.

Competitors: Andere Anbieter ähnlicher SECOs sind erfolgreicher und die Nutzer wechseln zu diesen. (Symbian und Windows Mobile vs. iOs und Android)

Regulatory Bodies: Microsoft wurde gezwungen eine Browserauswahl in Windows einzubauen.

Trade Associations: Wirtschaftsverbände schreiben ggf. (ethische) Richtlinien vor.

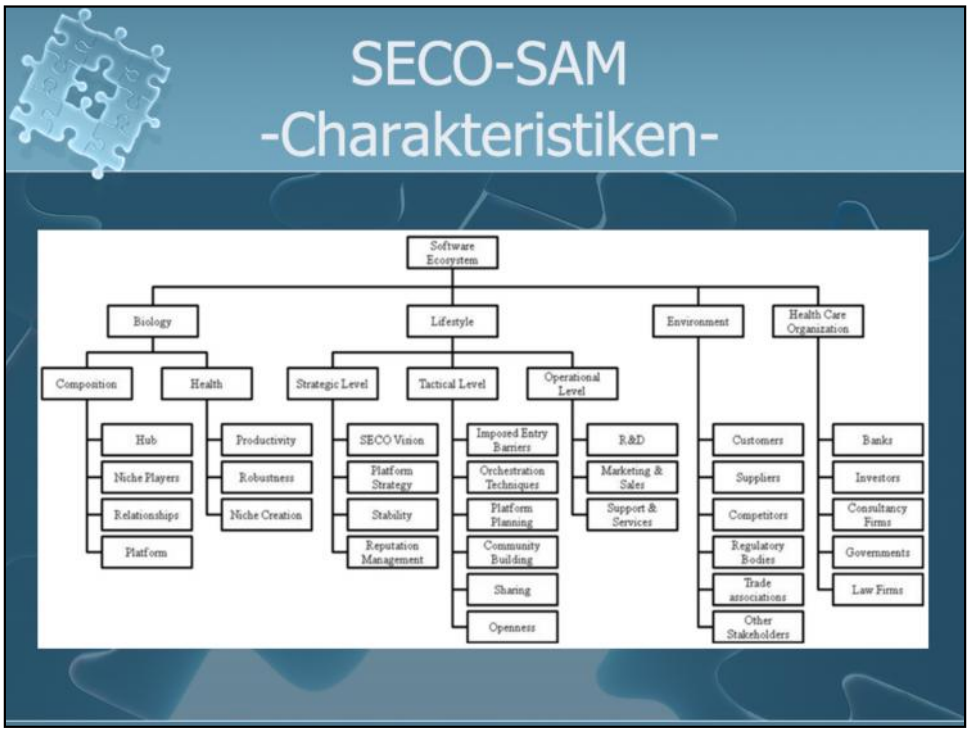
Other Stakeholders: Alle die in die Oberen Kategorien nicht eingeordnet werden können.

Banken und Investoren können Kapital liefern.

Regierungen können entweder Regulieren (siehe ‚Regulatory Bodies‘) oder auch helfen (z. B. Mit Subventionen). Deswegen werden sie hier aufgelistet.

Beratungsunternehmen helfen dem Anbieter eines SECOs die richtigen Strategien zu wählen.

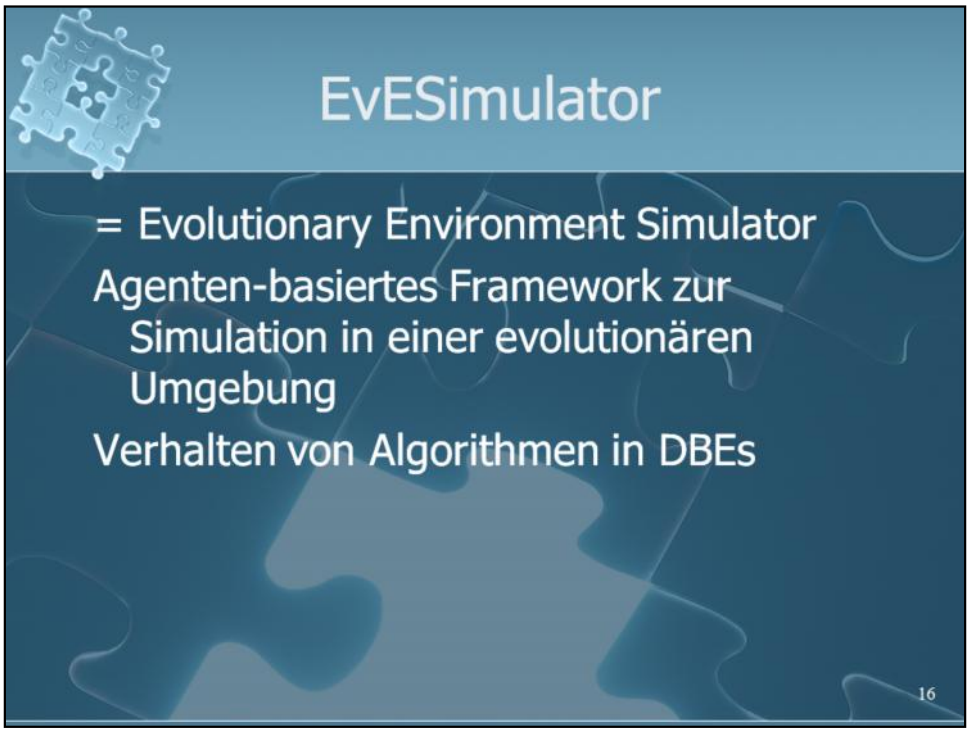
Anwaltskanzleien helfen bei rechtlichen Belangen.


SECO-SAM
-Charakteristiken-
Software Ecosystem
Biology
Lifestyle
Environment
Health Care Organization
Composition
Health
Strategic Level
Tactical Level
Operational Level
Hub
Niche Players
Relationships
Platform
Productivity
Robustness
Niche Creation
SECO Vision
Platform Strategy
Stability
Reputation Management
Imposed Entry Barriers
Orchestration Techniques
Platform Planning
Community Building
Sharing
Openness
R&D
Marketing & Sales
Support & Services
Customers
Suppliers
Competitors
Regulatory Bodies
Trade associations
Other Stakeholders
Banks
Investors
Consultancy Firms
Governments
Law Firms

Open Design Alliance:

Nonprofit-Organisation mit über 1200 Mitgliedern. Hub des SECO Teigha (development platform for creating engineering applications).


Agenda
Simulation
EvESimulator
Zielstellung
Modell
Ergebnisse
Smart Grid
SECO-SAM
AppEco
15



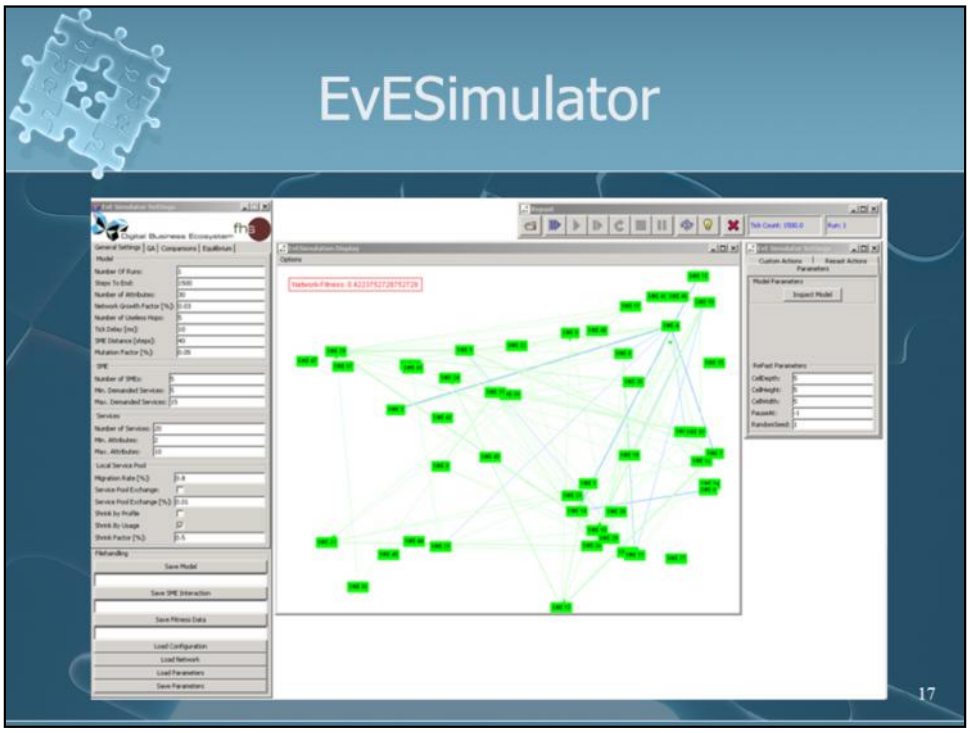
•EveSimulator = Evolutionary Environment Simulator

•DBE = Digital Business Ecosystem

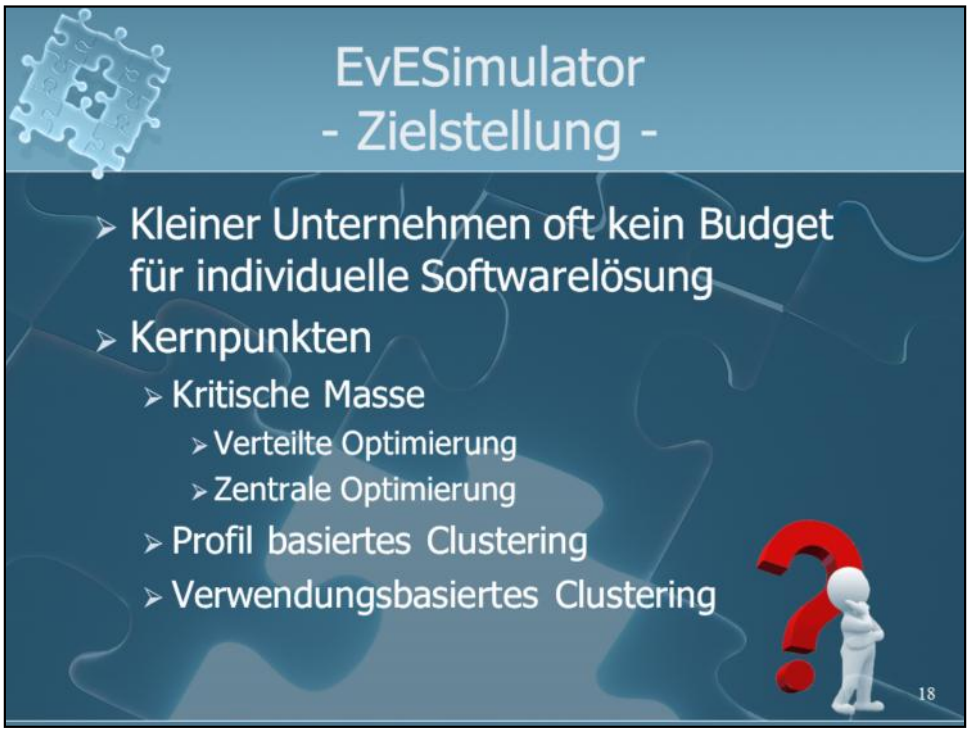
•Der EvESimulator ist ein agenten-basiertes Framework (jeder Knoten kann selbst agieren und Entscheidungen treffen).

•Der EvESimulator kann auch für die Simulation von Netzwerken allgemein verwendet werden. Er wird verwendet um das Verhalten verschiedener Algorithmen in DBEs zu untersuchen.

•EvESimulator dient der Simulation des Verhaltens einer Evolutionären Umgebung, damit Naturwissenschaftler, Sozialwissenschaftler, Wirtschaftswissenschaftler in einem Framework ihre Funde testen können.



Dies ist nur ein Beispiel-Screenshot von der EveSimulator Homepage.



•Kleinere Unternehmen verfügen meist nicht über das Budget für eine individuelle SW-Lösung. Daher wäre es ein Vorteil, würden sich die benötigten Dienste „selbst zusammenstellen“.

•Die Kernpunkte der Simulation sind:

- •Die Ermittlung der kritischen Masse, also dem Minimum von Individuen für den Erhalt eines effizienten Ecosystems.
  - zentral in Bootstrapping Phase besser, verteilt aber später
  - Anzahl Attribute & SMEs entscheidend für Ergebnisse
- •Funktioniert Clustering allgemein (über ähnliche Profile, ähnliche Anforderungen) durch Selbst-Optimierende DES?
- •Funktioniert Clustering basierend auf der Verwendung durch Selbst-Optimierende DES?

•Was vermutet ihr? (ca. 2 Minuten)

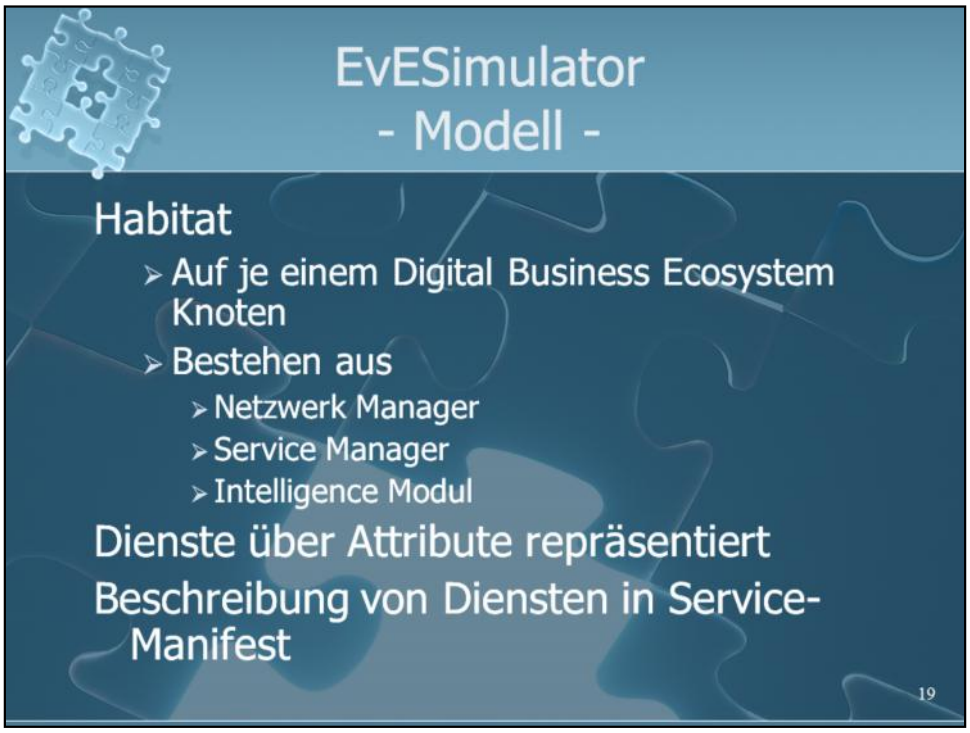
•Zentrale Optimierung: Zentrale Registry

•Verteilte Optimierung: Jeder Service Pool optimiert selbst, wobei Services je nach

vermuteter Verwendung in der Zukunft migrieren.

+ kleinerer Suchraum

- keine gesamt Kenntnis



• Habitat Knoten laufen auf DBE-Knoten (Digital Business Ecosystem).

•Netzwerk Manager

- •Der Netzwerk Manager ist zuständig für Discovery/Verbindung/Mapping der Aufrufe lokal zu Netzwerkressource Knoten.
- •Habitate sind über P2P verknüpft.

•Service Manager

- •Der Service Manager stellt die Funktionalität zur Verwaltung von Diensten. Diese liegen in einem Service Pool.
- •Habitate verwalten (löschen, speichern) Dienste und stellen sie anderen Habitaten zur Verfügung (Migration erfolgt über Kopieren des Dienstes), fall sie „erfolgreich“ sind.
- •Analyse-Ergebnisse sozialer Netzwerke werden als Basis um die Bereitschaft zum Informationsaustausch zu Messen verwendet, da Beziehungen und Vertrauen wichtig für die Dienstauswahl sind.
- •Dienste können nur bei einer existierenden Verbindung migrieren.

•Das Intelligence Modul besteht aus

- •Einem genetischen Algorithmus
- •Einer Fitness Funktion

•Dienste werden über Attribute repräsentiert (numerisch oder symbolisch (z. B. Farbe, können aber auch durch Numerische Werte repräsentiert werden)). In diesem Beispiel sind alle Attribute natürliche Zahlen

•Dienste Verfügen über ein Manifest mit einer Beschreibung des Dienstes etc. und über eine Historie.

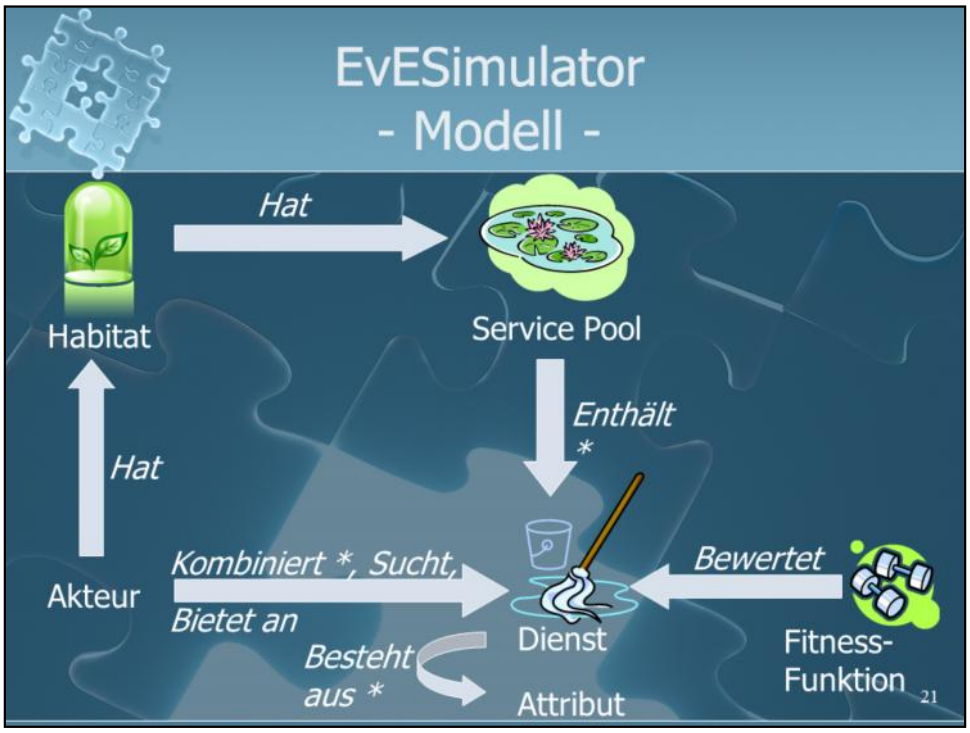
•Ein Akteur repräsentiert eine SME (Kleine und Mittelständische Unternehmen) innerhalb des verwendeten sozialen Netzwerks. Das Mapping Akteur, SME, Habitat ist hier immer 1 : 1. In der Simulation werden sie als Knoten dargstellt.

•Jeder Akteur kann verschiedene Dienste anbieten und/oder benötigen.

•Jeder Akteur hat ein Habitat mit Diensten, die er verwaltet. Dienste können virtuell oder real sein.

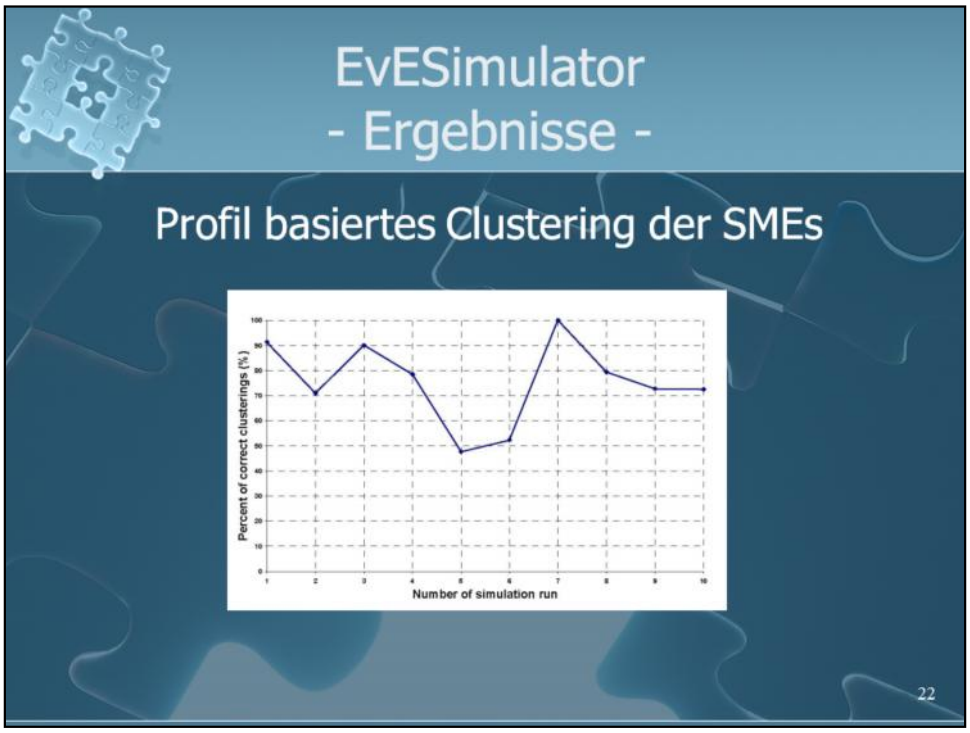
•Werden mehrere Dienste kombiniert bilden sie einen neuen Dienst mit der vereinigten Menge Attributen der einzelnen Dienste. Bsp. Ein Textverarbeitungsprogramm enthält Textverarbeitung, Rechtschreibprüfung und Auto-Vervollständigung.

•Neue Dienste erscheinen zuerst im Habitat der Teildienste, bzw. im Habitat des Akteurs, der die Dienste kombiniert.



•Der evolutionärer Algorithmus wählt über die Fitness-Funktion die besten Kombinationen für eine Anfrage aus. Der Algorithmus verwendet dafür den lokalen Service Pool.

•Die Fitness-Funktion bewertet Kombinationen von Diensten.



•SMEs sollten nach ähnlichen Anforderungen gruppiert werden.

•Hier sieht man die Auswertung zum Profil basierten Clustering. Je nach Iteration ist eine deutliche Schwankung in der Qualität des Clustering zu erkennen.

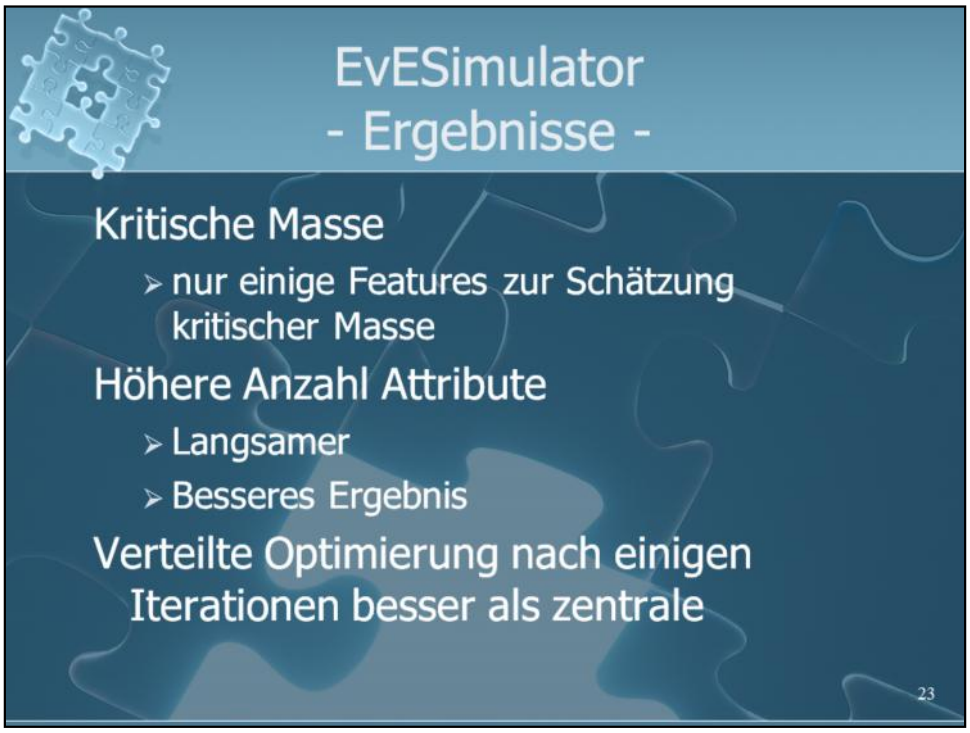
•Simulation mit zwei beabsichtigten Clustern:

- Bei über 90% der Iterationen von 10 Simulationen fand ein klares Clustering statt (die Dienste wurden also in die SMEs, in welchen sie benötigt werden, migriert).
- Aufgrund der großen Schwankungen des korrekten Clustering halte ich die Verwendbarkeit in der Praxis für fraglich (bei 5 Iterationen wurden unter 50% der SMEs korrekt geclustert).

•verwendungsbasiertes Clustering

- Es wurde die gleiche Simulation verwendet, nur die Einstellungen zum Management des Service Pools wurden geändert.
- Ergebnis: Wahrscheinlich nutzbar, die Cluster sind erkennbar.

•Aber: Nur vorläufige Experimente.



•Für die Auswertung zur Kritischen Masse wurden vier verschiedene Netzwerk mit unterschiedlichen Attributen und einer unterschiedlichen Anzahl an Attributen simuliert (10 SMEs jew. Mit 20, 40 und 50 Attributen und einmal 50 SMEs mit 20 Attributen).

•Der EvESimulator enthält nur einige Features zur Schätzung kritischer Masse an

•Es wurden zwei verschiedene Ansätze zur Optimierung simuliert:

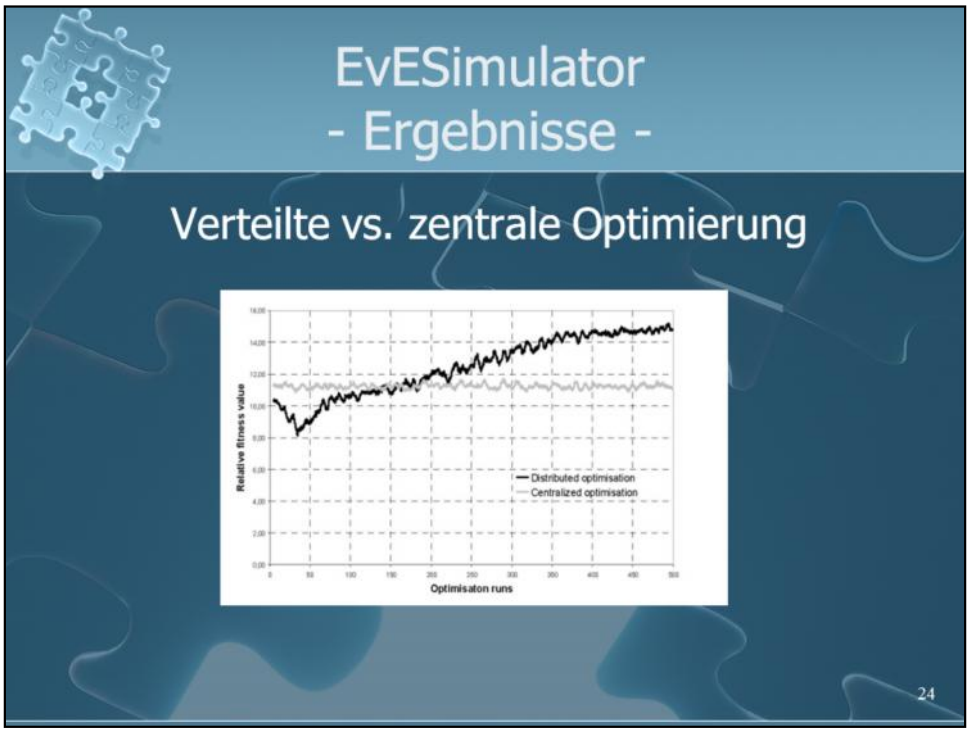
- Verteilte Optimierung

  besser in späteren Iterationen -> Bootstrapping Phase

- Zentrale Optimierung

  besser zu Beginn

•Die Simulation wird durch mehr Attribute zwar langsamer, die Ergebnisse aber besser.



•Hier seht ihr eine graphische Auswertung der Ergebnisse. Man sieht, dass zunächst die Verteilte Optimierung unterhalb der Zentralen verläuft, sie dann aber deutlich ansteigt.

•Nur bei mehr Attributen ist die verteilte Optimierung deutlich besser, nicht bei mehr SMEs.

•Begründung: Vermutet wird, das der Grund die Vorsortierung (lokaler Service Pool) ist, wodurch der Suchraum verkleinert wird.


Agenda
Simulation
SECO-SAM
EvESimulator
Smart Grid
Begriff
Modell
Simulation
AppEco
25

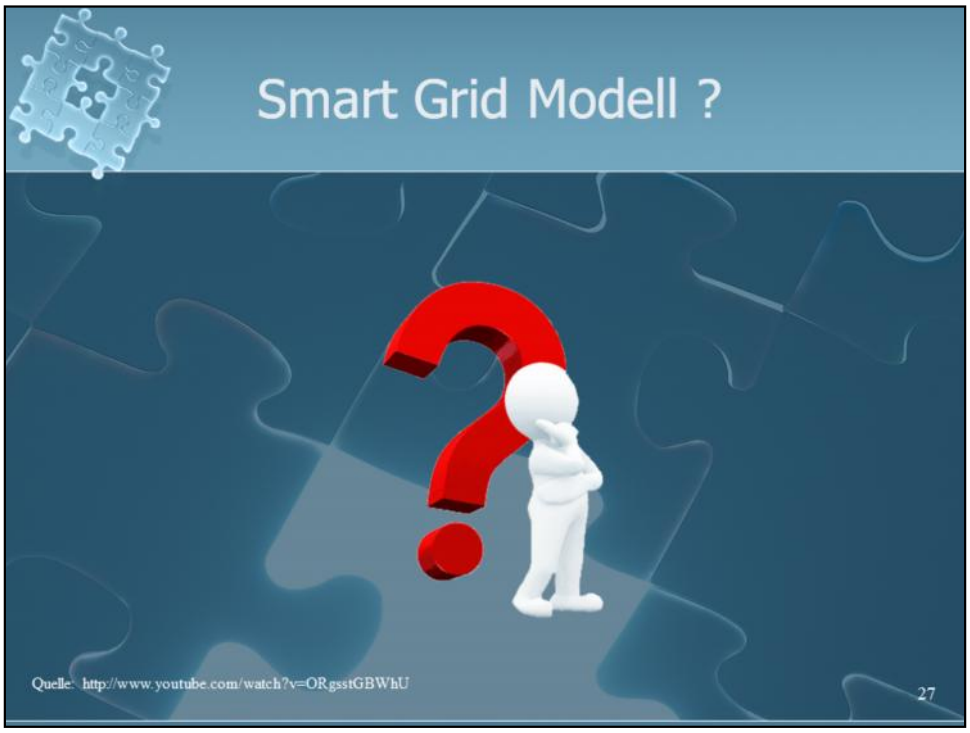
•PDF: 1-2011 d doppel S. 22 ff, 1-2012 D doppel S. 66 ff

•Dezentrale Energieerzeugungsanlagen sind neu, daher ist ihr Verhalten bei Lastsprünge und Einspeisesprünge noch unbekannt.

•Das Hauptproblem bei Dezentrale Energieerzeugungsanlagen sind inhomogene Erzeuger und Verbraucher. Es gibt einen zeitlichen Versatz zwischen Erzeugung und Bedarf (Bsp. Photovoltaik: Strom wird Tags produziert, aber Nachts die Beleuchtung genutzt).

•Regulierende Elemente sollen diese Inhomogenität kompensieren : z. B. Speicher oder Motoren, die Magnetfelder erzeugen (Strom wird bezogen und wieder zurück ins Netz gespeist, aber mit entsprechender Herz-Zahl etc.).

•Der bisherige Aufbau des Stromnetzes war Hochspannungsnetz -> Mittelspannungsnetz -> Niederspannunsgnetz -> Endverbraucher.

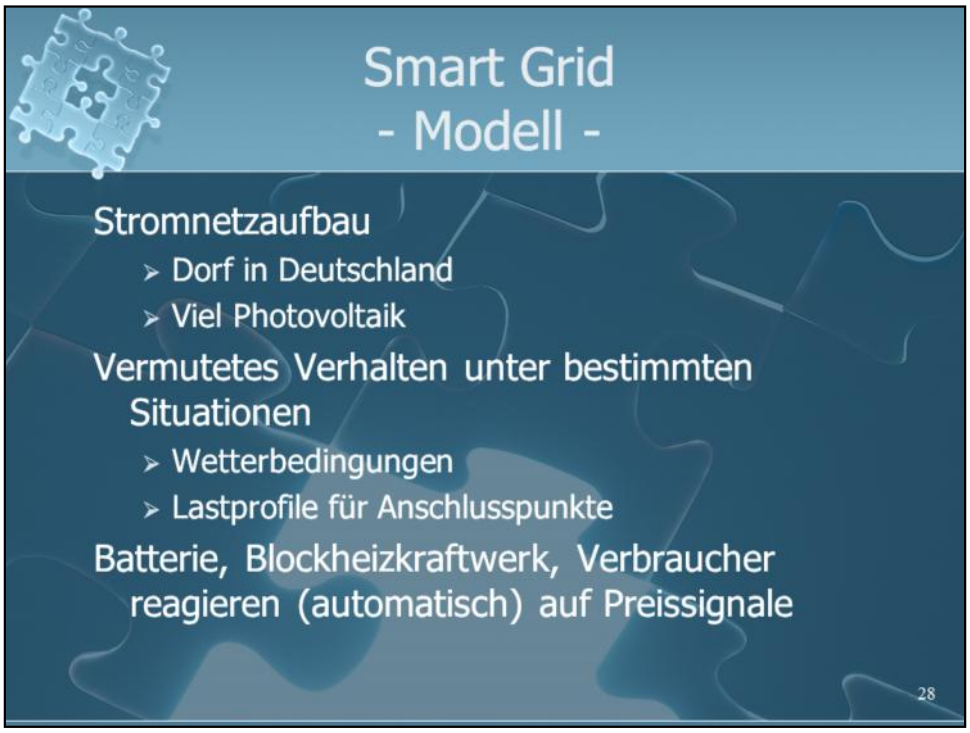


•Entsprechend werden diese bei den Simulationen berücksichtigt.

•Da eine Simulation aber immer eine Abstraktion als Modell verwendet, werden nie alle Faktoren berücksichtigt.

•Erst den Clip von http://www.youtube.com/watch?v=ORgsstGBWhU. (Dieser fehlt hier im Moment, da der Email-Anhang sonst zu groß würde.)

•Ca. 2 Minuten: Was braucht man zur Simulation bzw. für das Modell eines Smart Grids? Welche Faktoren müssen berücksichtigt werden?



-Smart Grid Simulation von Siemens

-Stromnetzaufbau: Z. B. eine Stadt und ihre “Zulieferer”, Photovoltaik, Windkraft etc.

Beispiel : Verhältnisse im Stromnetz eines realen Dorfes in Deutschland mit hohem Anteil Photovoltaik,

•Batterie, Blockheizkraftwerk und Verbraucher reagieren auf Preissignale: Der Preis steigt, wenn wenig Solarenergie erzeugt wird, dadurch wird das Kraftwerk zugeschaltet.

•Die Verbraucher reagieren auf Preissignale, d. h. der Verbrauch kann sinken. Hintergedanke: „Zu teuer, also lass ich die Klimaanlage aus“ oder die Waschmaschine läuft immer wenn der Preis einen bestimmten Wert hat usw. (Siemens entwickelt in Kooperation mit der TU München bereits Software-Werkzeuge, mit denen sich die Gebäudetechnik künftig nach dem

Stromangebot steuern lässt)

-Wetterbedingungen, Lastprofile für Anschlusspunkte sind variable und werden im Verlauf der Simulation verändert

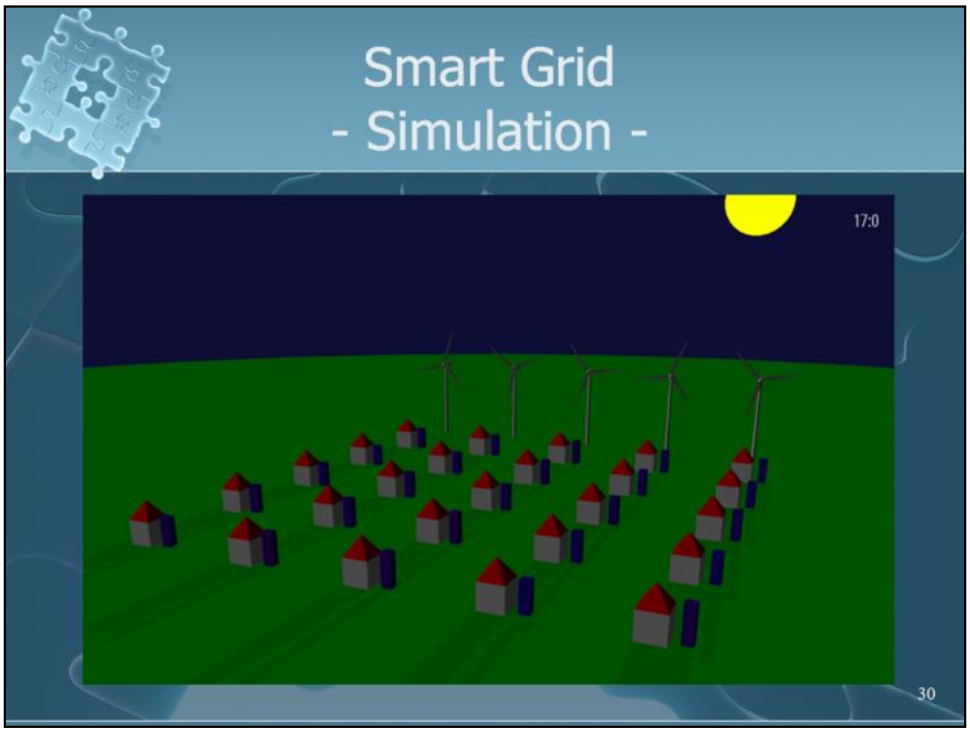
-Lastpofile beziehen sich auf einzelne Haushalte / Gewerbebetriebe, nicht einzelne Geräte.

-Die Kommunikation der Last erfolgt über intelligente Zähler (heutige moderne Stromzähler).

⇒Bei einer Simulation mit viel Sonnenschein viel ein Straßenzug mit viel Photovoltaik auf, in diesem kommt das Stromnetz an seine Belastungsgrenze.

⇒Zur Überprüfung und weiteren Analyse wurde das Dorf im Labor (in verkleinertem Maßstab) nachgebaut.

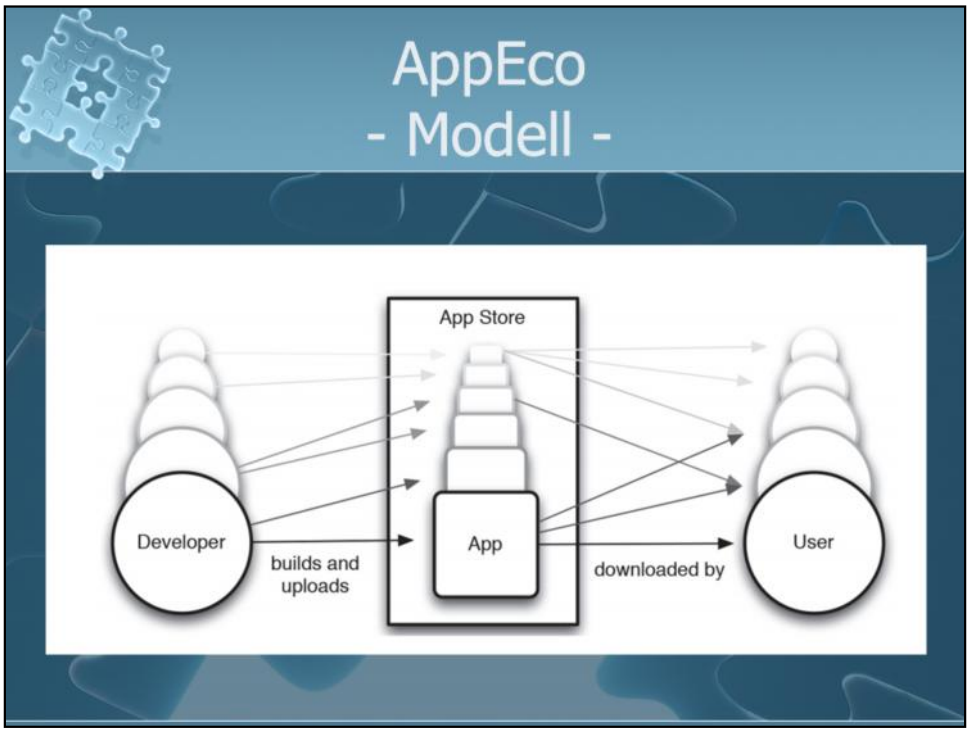
⇒Bei einer anderen Simulation war der Himmel bewölkt. Daher wurde zu wenig Strom erzeugt und automatisch die Batterie zugeschaltet, bis das Blockheizkraftwerk hochgefahren wurde.



Dies ist ein Screenshot des Simulationstool vom „Lab: Virtual Smart Grid Testbed“ der Smart Grid Research Group der TU-München.


Agenda
Simulation
SECO-SAM
EvESimulator
Smart Grid
AppEco
Modell
Kalibrierung
Simulation
Ergebnisse
31


AppEco
Simuliert einen App Store
Verschiedene Strategien für App Entwickler
Welche Strategie führt zu meisten App-Verkäufen
Welche Strategie ist am besten für das Ecosystem des App-Stores


AppEco
- Modell -
App Store
Developer
builds and
uploads
App
downloaded by
User

Zusätzlich hat ein Developer noch eine Entwicklungsgeschwindikeit für Apps.

Zur Ermittlung realitischer Werte werden echte App Stores und deren Entwickler betrachtet.

Typen

-Innovator

Generieren zufällige App für jeden Durchgang

-Milker

Generieren eine zufällige App im ersten Schritt. Diese wird in jeder Iteration leicht verändern.

-Optimiser

Generieren eine zufällige App. In jeder Iteration ermitteln sie die erfolgreichste ihrer Apps und verändern diese leicht. (ähnlich einer hill climbing Optimierung)

-Copycat

Sucht sich eine App aus den best verkauften Apps des Appstores und verändert diese leicht.

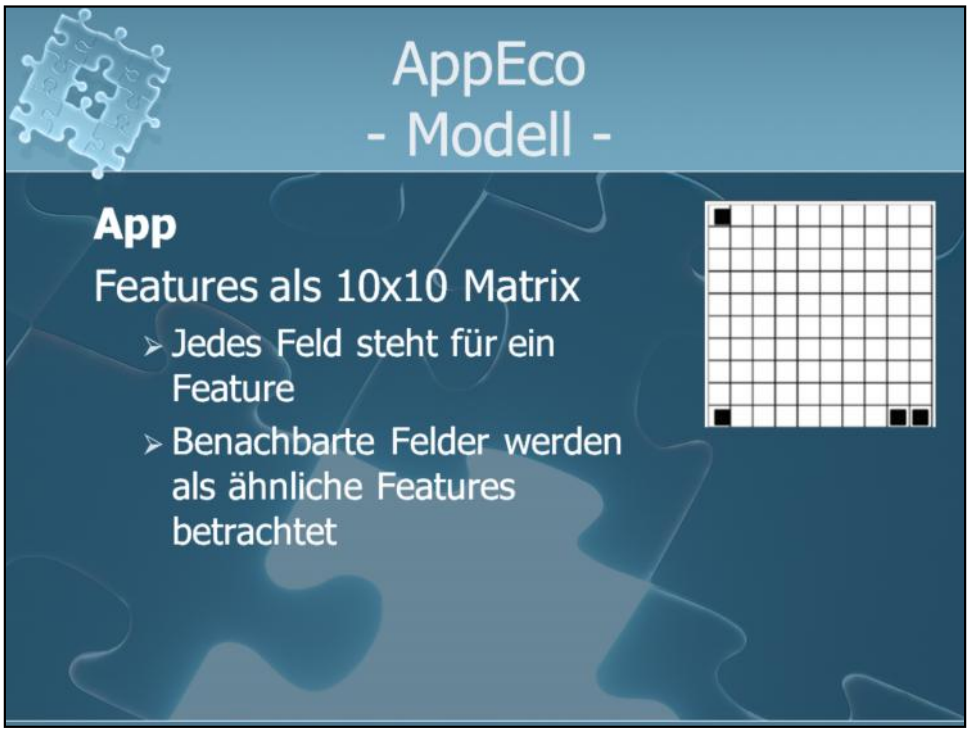
-Flexible

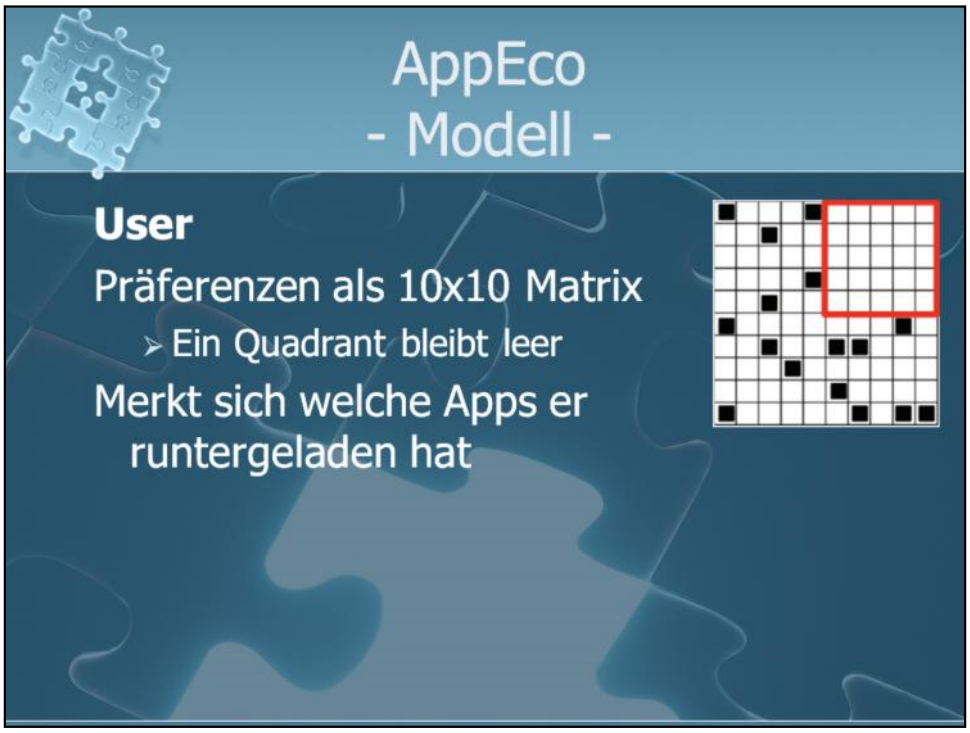
Ändert seine Strategie entsprechend der erfolgreich Apps

App Store hat Parameter die bestimmen wie viele Apps in den einzelnen Listen zu finden sind.

Die Liste der Apps mit den meisten Downloads bezieht sich auf Downloads der letzten Tage, wobei die Tage unterschiedlich gewichtet werden, je nachdem wie lange sie her sind werden sie weniger gewichtet.

Keywordsuche gibt aus Gründen der Vereinfachung nur eine Liste von zufällig ausgewählten Apps zurück.


AppEco
- Modell -
App
Features als 10x10 Matrix
Jedes Feld steht für ein Feature
Benachbarte Felder werden als ähnliche Features betrachtet



Leerer Quadrant stellt nicht erwünschte ‚Feature' dar. (negative Eigenschaften)

Developer kennen die Präferenzen der Benutzer nicht.

Jeder User merkt sich welche Apps er/sie gedownloadet hat.

Außerdem merkt sich jeder User wie lange es her ist, dass er/sie eine App gesucht hat. Dazu noch eine minimal und maximal Zeit die er wartet bis er/sie wieder nach einer App sucht.

Pop*minUser*/**Pop***maxUser*:Anfangs- und Endanzahl an Usern

D*user*: Koeffizient für Userzahlanstieg

S*user*: Koeffizient für Userzahlanstieg

Pop*minDev*/**Pop***maxDev*: Anfangs- und Endanzahl an Developern

D*dev*: Koeffizient für Developerzahlanstieg

S*dev*: Koeffizient für Developerzahlanstieg

N*InitApp*: Anzahl der Apps im App Store bevor die User sich Apps runterladen können

Bro*min*/ **Bro***max*: Tage die die User warten bevor sie wieder nach Apps suchen

Dev*min*/**Dev***max*: Tage für Entwicklung einer App

P*pref*: Wahrscheinlichkeit das ein Feld in den Präferenzen eines Users ausgefüllt ist.

P*feat*: Wahrscheinlichkeit das ein Feld in den Features einer App ausgefüllt ist.

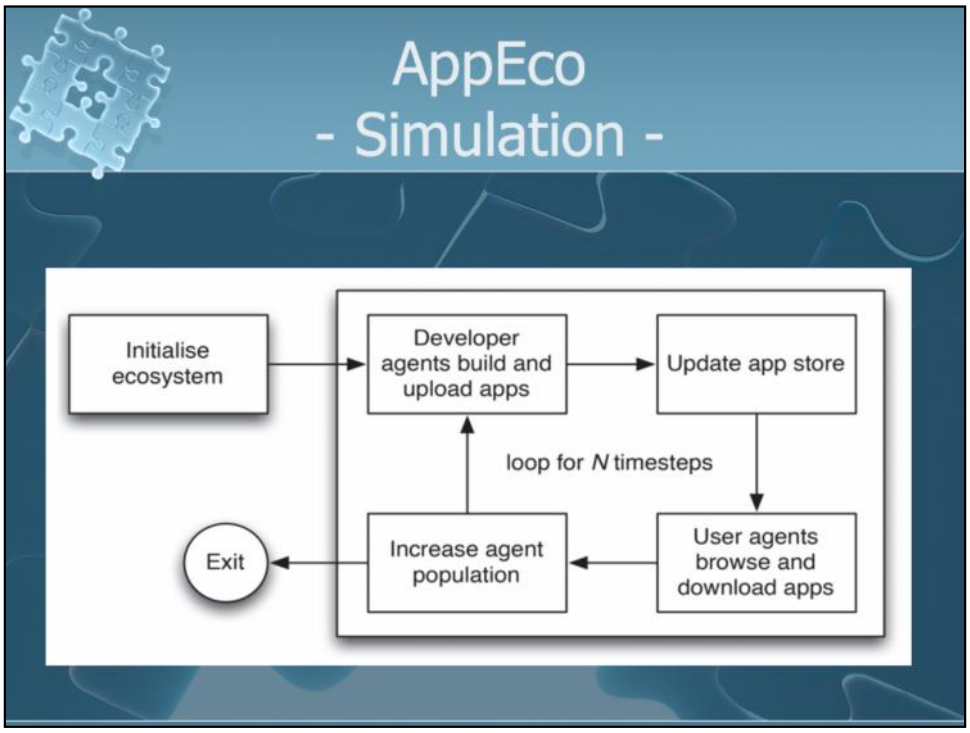
P*OnNewChart*: Wahrscheinlichkeit das eine App in der Liste der neuen Apps auftaucht.

N*MaxNewChart*: Maximale Anzahl der Apps in der Liste der neuen Apps.

N*MaxTopChart*: Anzahl der Apps in der der Topliste

P*inactive*: Wahrscheinlichkeit das ein Developer aufhört Apps zu entwickeln.

Key*min*/**Key***max*: Minimal und Maximal Anzahl der Apps die ein User bei einer Keyword-Suche angezeigt bekommt.


AppEco
- Simulation -
Initialise ecosystem
Developer agents build and upload apps
Update app store
loop for N timesteps
User agents browse and download apps
Increase agent population
Exit

	AvgDl	Top20TotDl	Top20AvgDl	ZeroDl	FeatCV
S0 Innovator	1.18 (0.14)	3.85% (4.20%)	9.10% (5.52%)	26.51% (0.23%)	**1.40% (0.13%)**
S1 Milker	1.19 (0.14)	5.95% (5.16%)	8.80% (6.52%)	32.85% (0.26%)	4.27% (0.35%)
S2 Optimiser	1.41 (0.15)	7.40% (6.05%)	9.25% (7.30%)	32.90% (0.27%)	6.50% (0.62%)
S3 Copycat	**8.22 (0.41)**	**82.80% (8.30%)**	**72.85% (10.03%)**	**7.74% (0.23%)**	54.36% (6.78%)

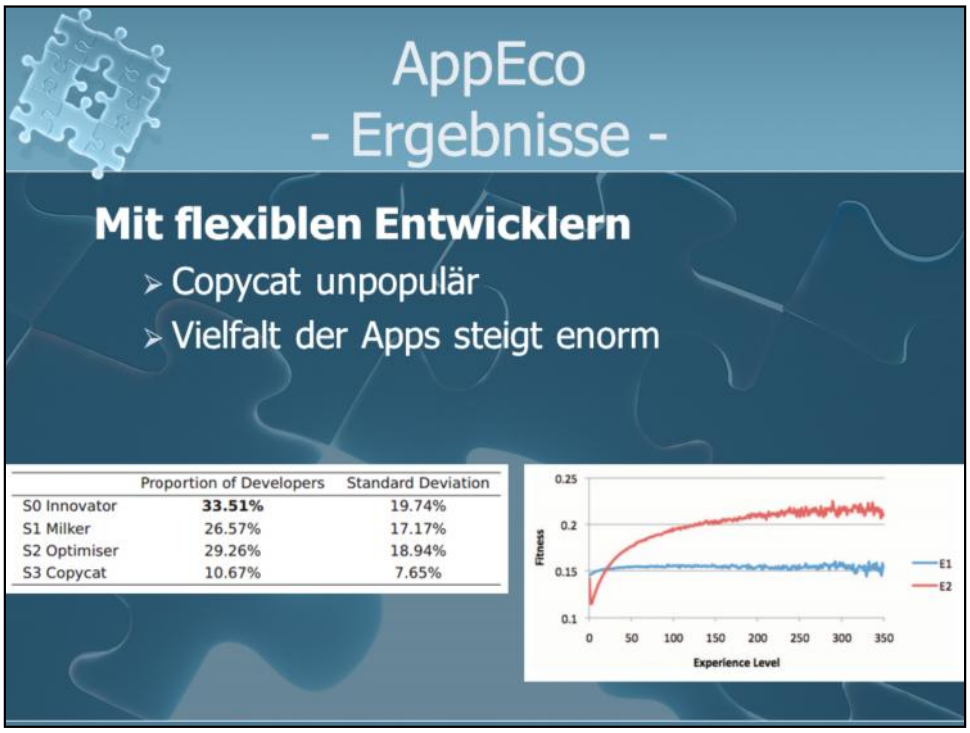
AvgDl: Durschnittliche Downloadzahl der Apps der Developer dieses Typens (in Tausend)

Top20TotDl: Anteil von Developern in den Top20 vom jeweiligen Typ bei dem meisten gesamten Downloads

Top20AvgDl: Anteil von Developern in den Top20 vom jeweiligen Typ bei dem meisten durchschnittlichen Downloads pro Apps

ZeroDl: Anteil an Developern von denen gar keine Apps runtergeladen wurden

FeatCV: Ein Wert der versucht auszudrücken wie sehr die Apps des Developers die Präferenzen aller Nutzer zu erfüllen (niedrige Werte sind als besser für die Gesundheit des SECOs zu betrachten)

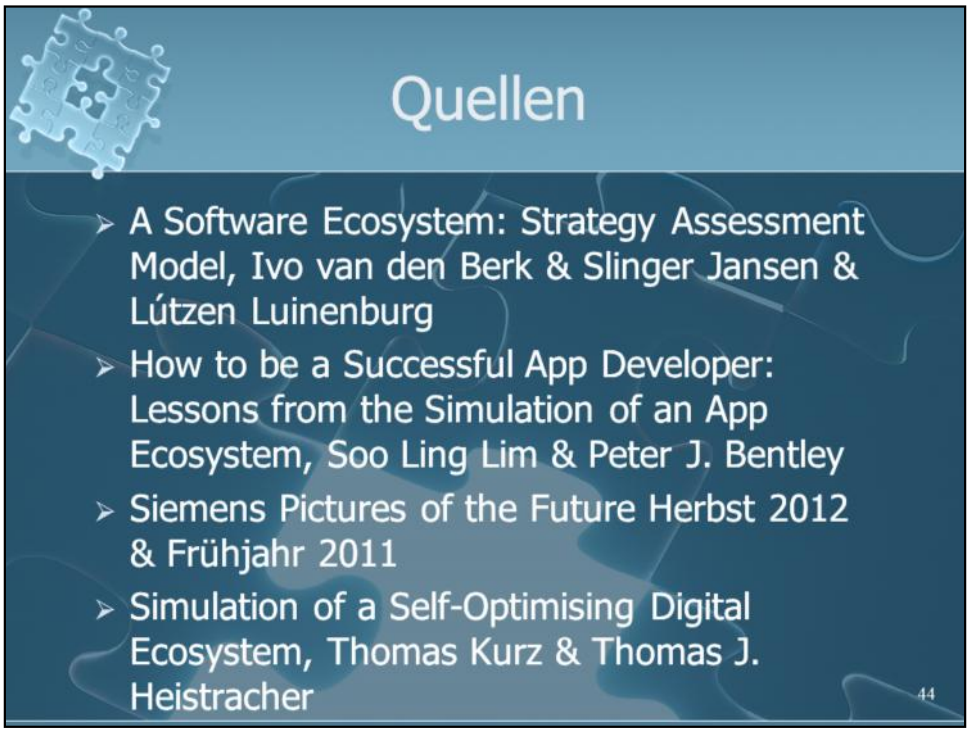


Fest: FeatCV 6.28% (Standard Abweichung 0.72%)

Flexibel: FeatCV 2.87% (Standard Abweichung 2.33%)


Vielen Dank für ihre Aufmerksamkeit!
Fragen?
42


Bildquellen
http://jura27.wordpress.com/2012/07/21/bin-ich-wirklich-eine-masterzahl/fragezeichen-2/
http://www.automationfederation.org/graphics/af/Smart-Grid.jpg
http://smartgrid.in.tum.de/activities/2012_winter_lab_testbed/
http://www.colourbox.de/preview/1845635-153605-r-bauplan-fur-ein-haus.jpg
43


Quellen
A Software Ecosystem: Strategy Assessment Model, Ivo van den Berk & Slinger Jansen & Lútzen Luinenburg
How to be a Successful App Developer: Lessons from the Simulation of an App Ecosystem, Soo Ling Lim & Peter J. Bentley
Siemens Pictures of the Future Herbst 2012 & Frühjahr 2011
Simulation of a Self-Optimising Digital Ecosystem, Thomas Kurz & Thomas J. Heistracher
44