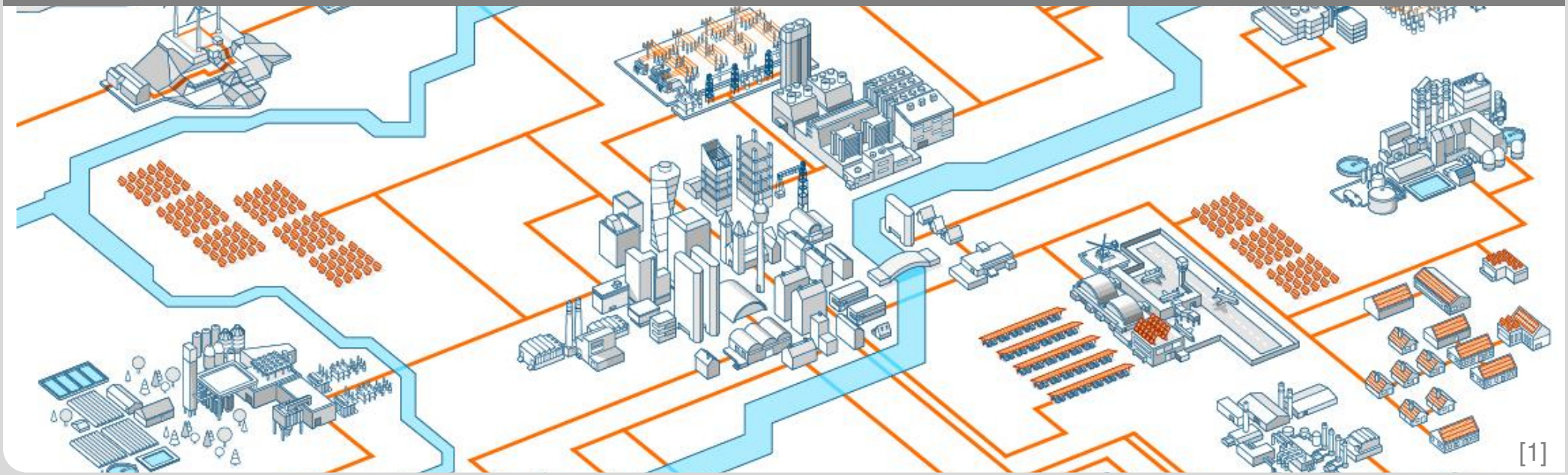
Karlsruher Institut für Technologie

Dezentrale Optimierung in Smart Grids

Vortrag · 09. Januar 2018

Moritz Winter
Betreuer: Ingo Mauser

SEMINAR ENERGIEINFORMATIK



[1]

Übersicht

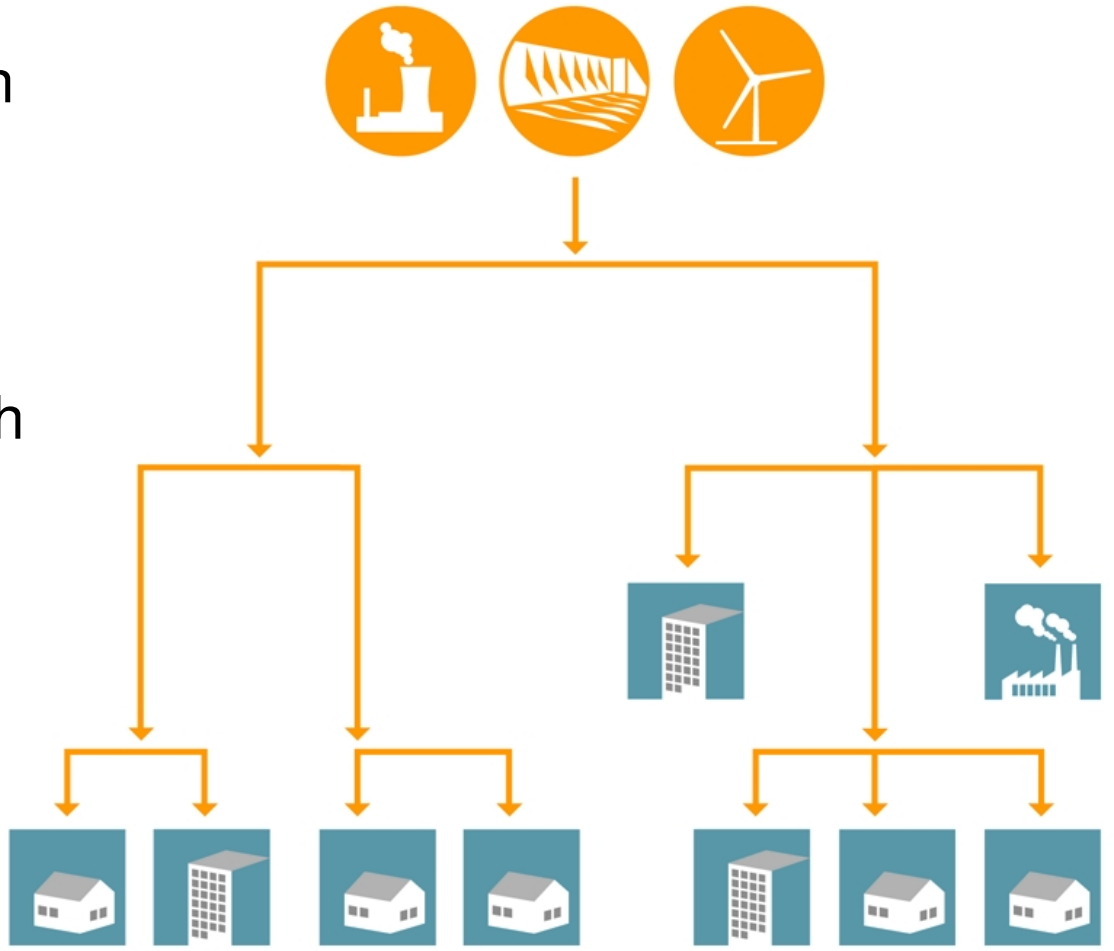
- Einführung: Begriffe
- Agent-basierte Netze
- COHDA: Dezentrale, selbstorganisierte Fahrplangenerierung
- Blockchain
  - Was ist das?
  - Beispiele: Bitcoin & Ethereum
  - Smart Contracts
  - ADMM mit Blockchain

Einführung – Begriffe: zentale Energienetze

- **Zentral:** Zentrale Autorität steuert Geschehen

Einführung – Begriffe: zentale Energienetze

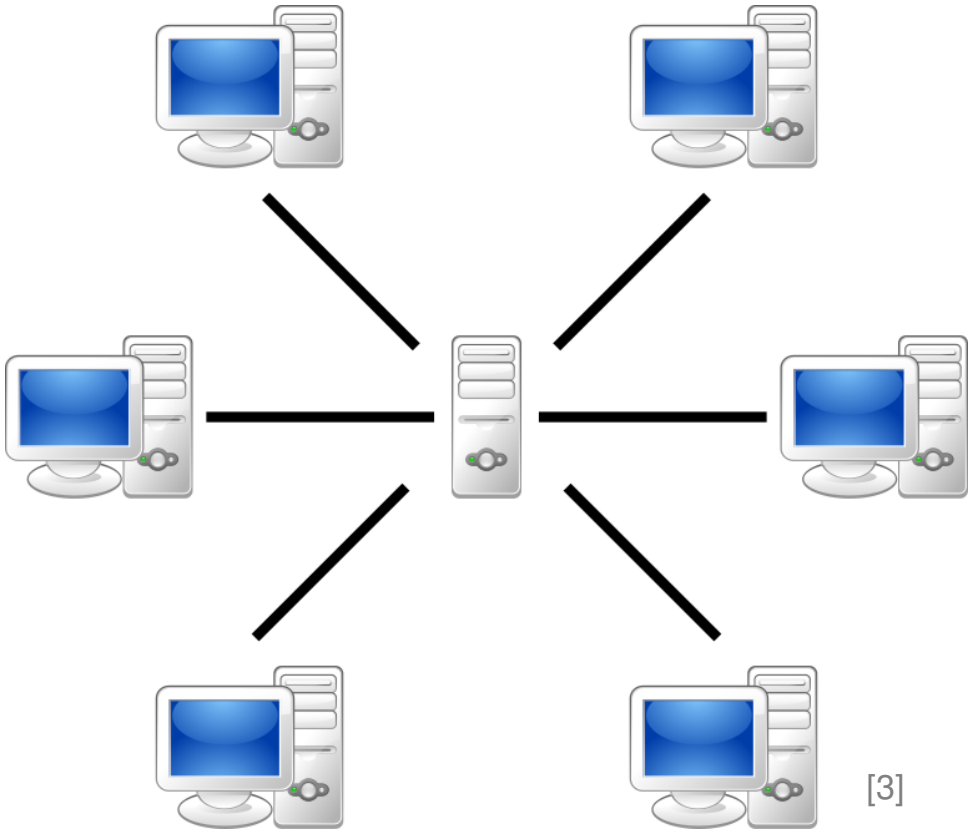
- **Zentral:** Zentrale Autorität steuert Geschehen
- Zentrales Energienetz
  - Große Kraftwerke speisen Energie ein
  - Angebunden auf Übertragungsnetzebene
  - Erzeugung folgt Verbrauch



[2]

Einführung – Begriffe: Zentrale Rechnernetze

- **Zentral:** Zentrale Autorität steuert Geschehen
- Zentrales Rechnernetz (Overlay)
  - Rechner kommunizieren über vorhandene Infrastruktur (Internet)
  - Zentraler Server stellt Dienst bereit
  - ”Single point of failure”



[3]

Einführung – Begriffe: Verteilte Stromerzeugung

- **Verteilt:** Geographisch verteilte Teilnehmer
  - Kann auch zentral gesteuert sein
  - Ressourcen sind verteilt

Einführung – Begriffe: Verteilte Stromerzeugung

- **Verteilt:** Geographisch verteilte Teilnehmer
  - Kann auch zentral gesteuert sein
  - Ressourcen sind verteilt
- Verteilte Erzeugung im Stromnetz
  - Immer mehr räumlich verteilte Erzeuger
  - Verbrauch nah am Erzeuger: Vermeidung von Netzverlusten
  - Statt alles über das zentrale Netz zu verteilen, möglichst lokal Erzeuger/Verbraucher verbinden

Einführung – Begriffe: Verteilte Rechnernetze

- **Verteilt:** Keine geographische Mitte
  - Kann auch zentral gesteuert sein
  - Ressourcen sind verteilt
- Verteilte Rechnernetze (Overlay)
  - Beispiel: Distributed-Hashtable (DHT): Daten werden verteilt und redundant im Netz gespeichert
  - Beispiel: Verteiltes Rechnen

Einführung – Begriffe: Dezentrale Energienetze

- **Dezentral:** Keine zentrale Autorität
  - Entscheidungen werden lokal getroffen
  - Lokale Ziele können berücksichtigt werden

Einführung – Begriffe: Dezentrale Energienetze

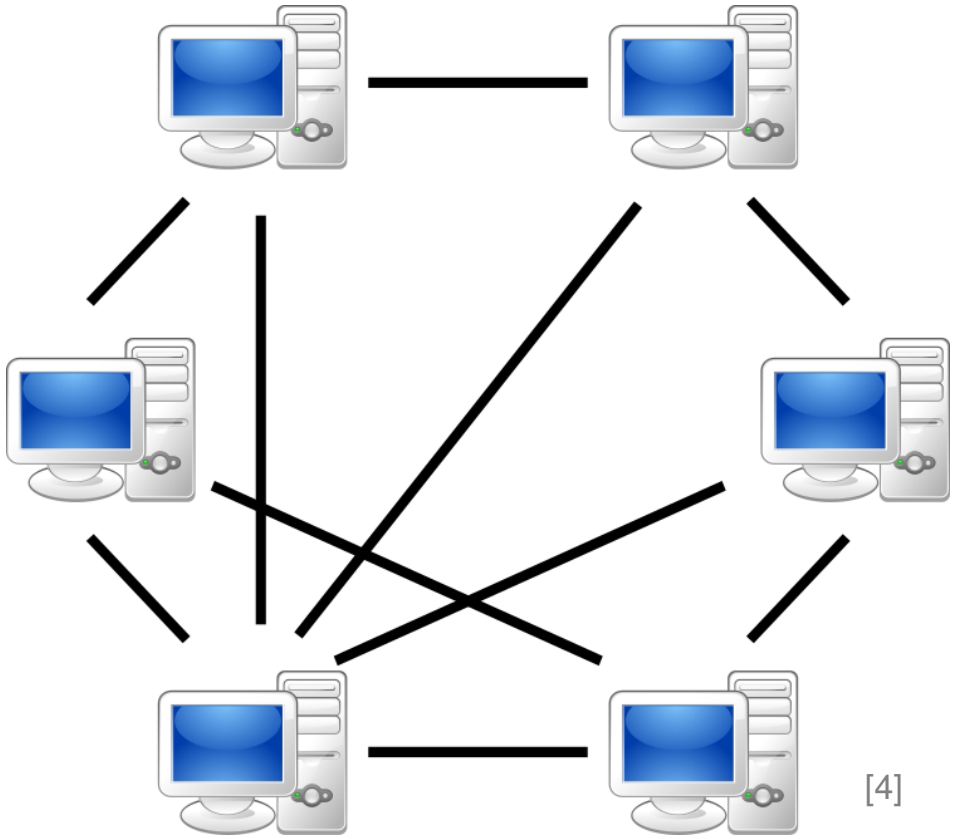
- **Dezentral:** Keine zentrale Autorität
  - Entscheidungen werden lokal getroffen
  - Lokale Ziele können berücksichtigt werden
- Dezentrale Energienetze
  - Bündelung von verteilten Erzeugern/Verbrauchern in Virtual-Power-Plants (VPPs) oder Microgrids
  - Steuerung komplett dezentral oder hierarchisch

[2]

Einführung – Begriffe: Dezentrale Rechnernetze

- **Dezentral:** Keine zentrale Autorität
  - Entscheidungen werden lokal getroffen
  - Lokale Ziele können berücksichtigt werden
- Dezentrale Rechnernetze (Overlay)
  - Jeder Client gleichzeitig Server
  - Indirekte Kommunikation
  - Selbstorganisation

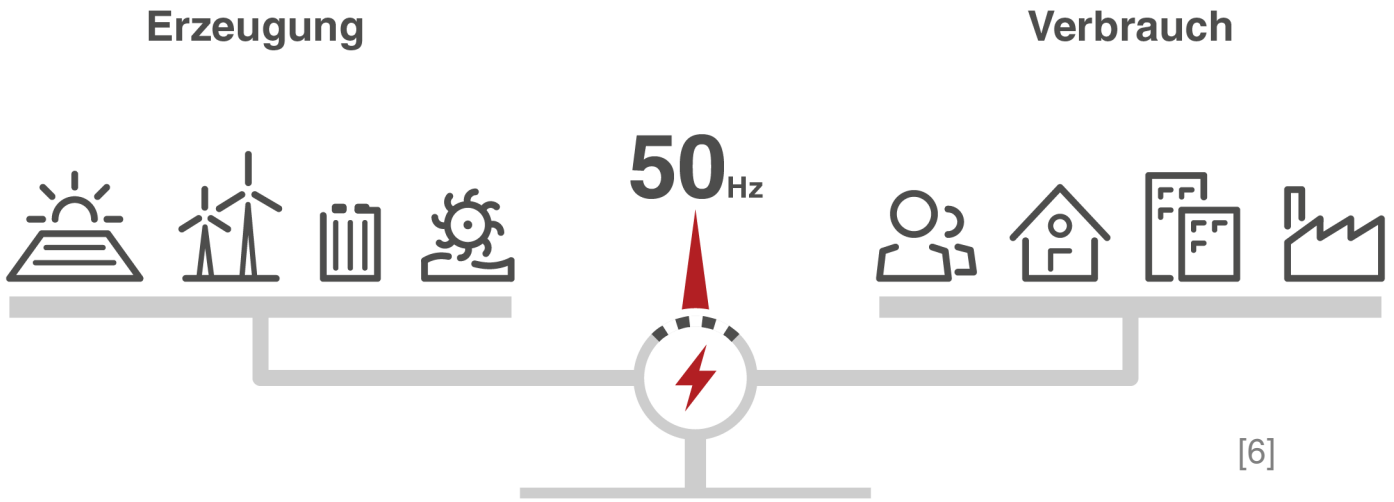
  → Peer-to-Peer Netz (P2P)



[4]

Einführung – Flexibilität

- Klassisch: Erzeugung an Verbrauch anpassen
- Neu: Verbrauch an Erzeugung anpassen
- Erzeuger/Verbraucher verfügen über Flexibilität die Koordiniert werden muss



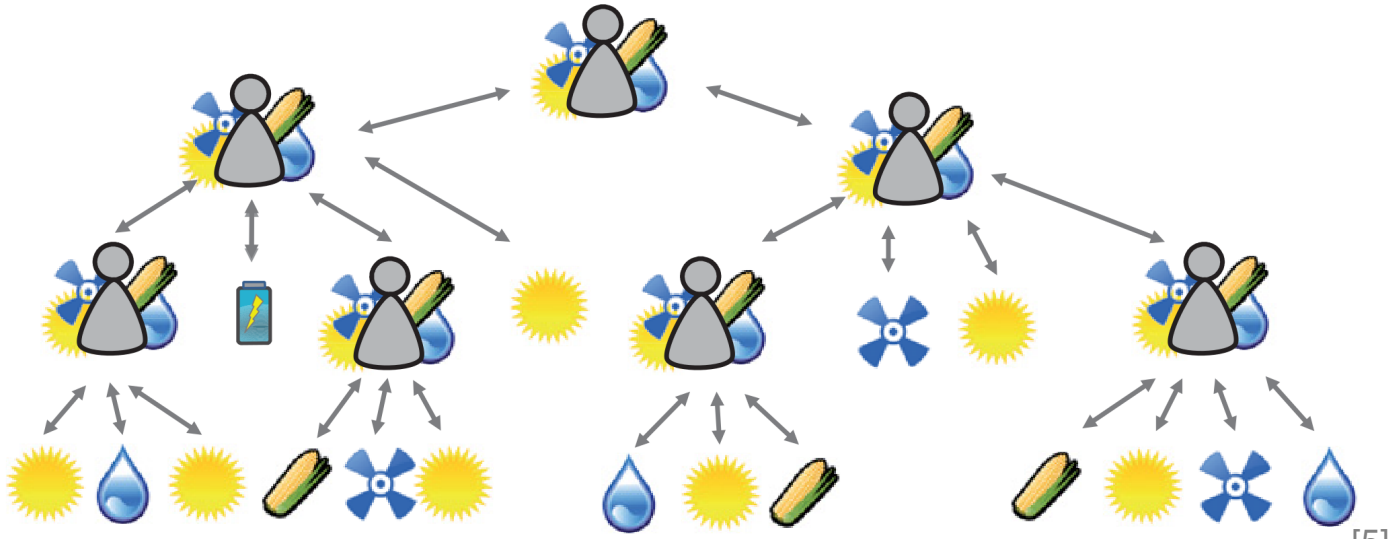
[6]

Einschub – Virtual Power Plant (VPP)

- Organisationseinheit für mehrere Erzeuger/Verbraucher
- Statisch oder dynamisch
- Kontrolliert durch ein Energiemanagement-System (EMS)
  - Muss Fahrplan erfüllen
  - Technische/Ökonomische Restriktionen der Erzeuger/Verbraucher müssen beachtet werden
- Einheit auf dem Energiemarkt: ökonomisch konkurierend zu traditionellen Kraftwerken
- Kann im Gegensatz zu **Microgrids** auch verteilt sein

Hierarchisches Stromnetz

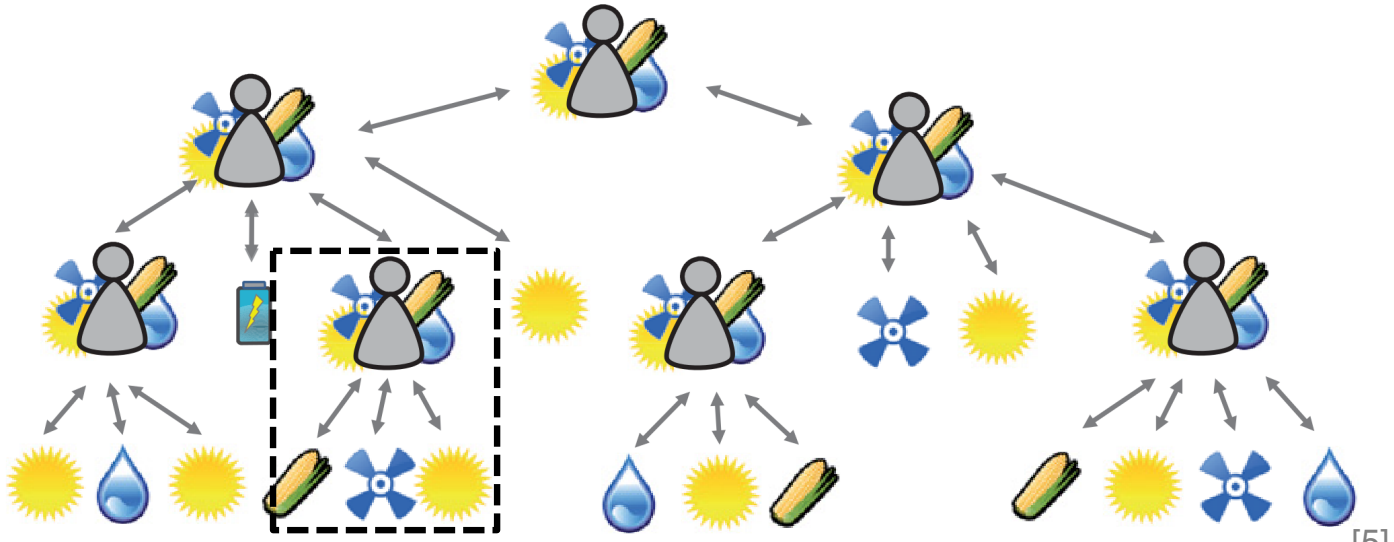
- **Hierarchie** von VPPs: Mischform **zentral/dezentral**
- Ungleichgewicht lokal ausgleichen: Ausbreitung durch Netz vermeiden
- Hierarchische Struktur kann zur Laufzeit angepasst werden (dynamisches VPP)
- Möglichst heterogene Struktur um Robustheit zu erhöhen



[5]

Hierarchisches Stromnetz

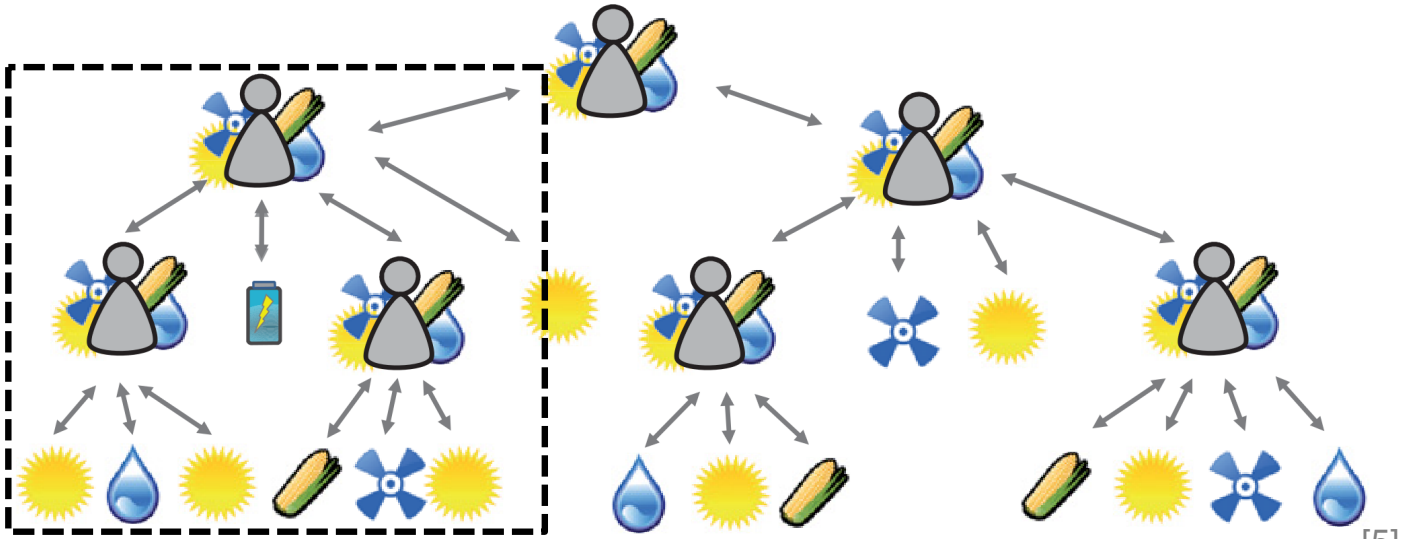
- **Hierarchie** von VPPs: Mischform **zentral/dezentral**
- Ungleichgewicht lokal ausgleichen: Ausbreitung durch Netz vermeiden
- Hierarchische Struktur kann zur Laufzeit angepasst werden (dynamisches VPP)
- Möglichst heterogene Struktur um Robustheit zu erhöhen



[5]

Hierarchisches Stromnetz

- **Hierarchie** von VPPs: Mischform **zentral/dezentral**
- Ungleichgewicht lokal ausgleichen: Ausbreitung durch Netz vermeiden
- Hierarchische Struktur kann zur Laufzeit angepasst werden (dynamisches VPP)
- Möglichst heterogene Struktur um Robustheit zu erhöhen

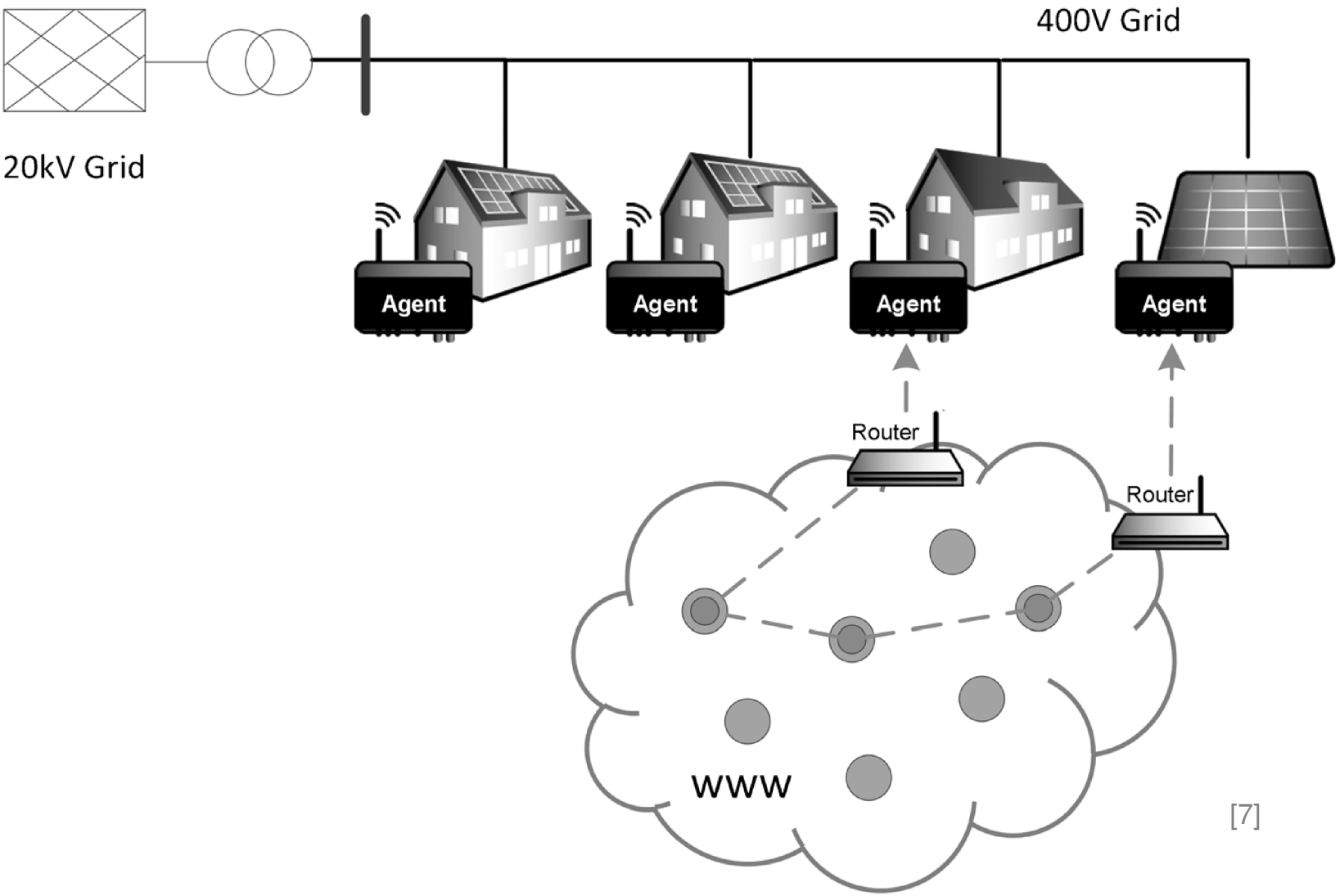


[5]

Agent-basierte Netze (1)

- Design von Gregor Rohbogner et al. [RFB14]
- Multi-Agent-System
- Ein Agent ist ein Softwareartefakt mit Kontrolle über ein oder mehrere Erzeuger/Verbraucher
- Agenten sollen:
  - Optimieren: Beste Lösung für Problem finden
  - Kontrollieren: Verantwortlich für technisches System
  - Lernen: Umgebungsänderungen adaptieren
  - Kommunizieren: z.B. mit anderen Agenten

Agent-basierte Netze (1)

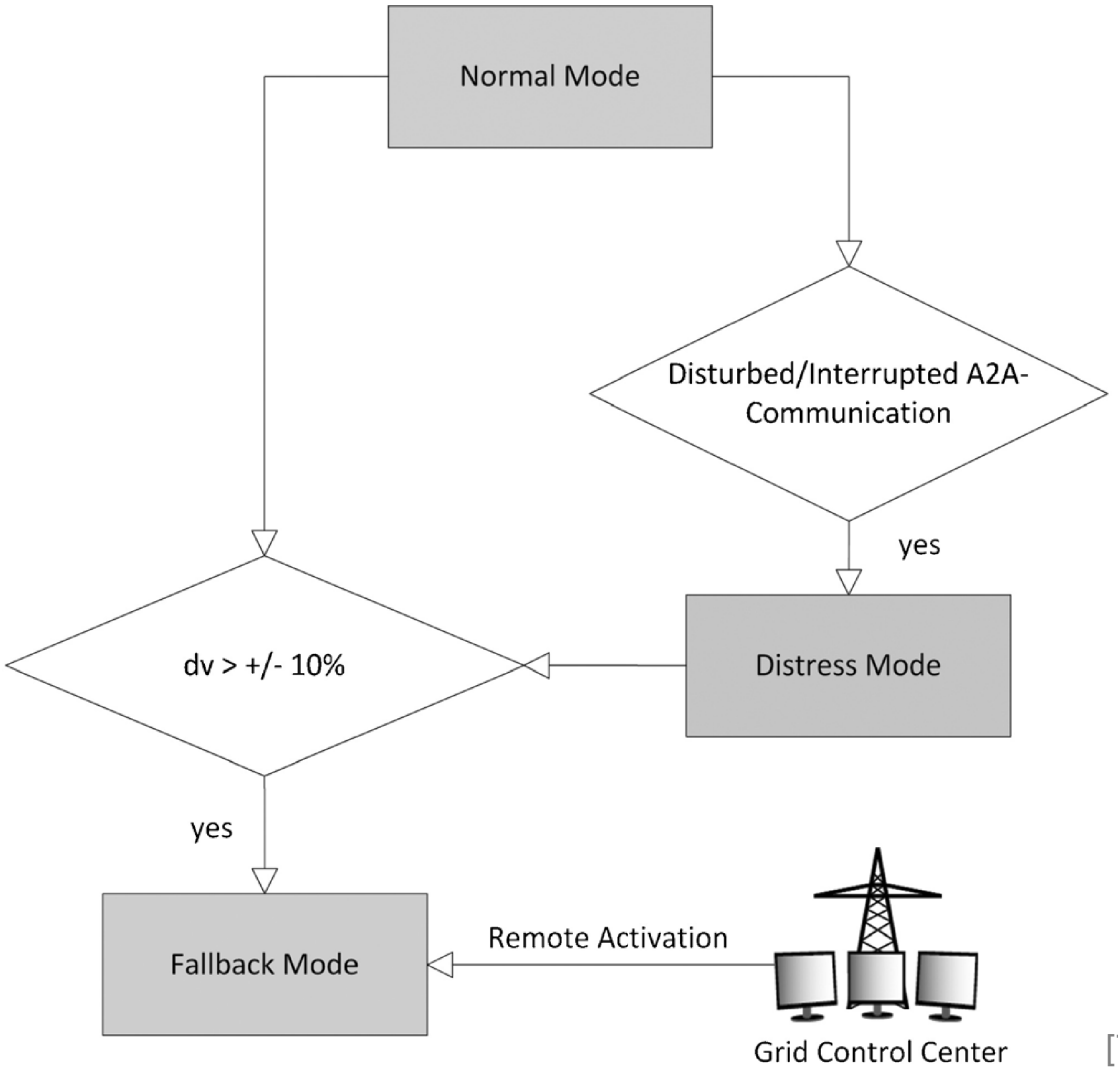


[7]

Agent-basierte Netze (2) – Kommunikation

- Dezentrale Agent zu Agent Kommunikation (A2A)
  - Über LAN ans Internet angebunden
  - Transparente P2P-Kommunikation mit anderen Agents
    - *Session Traversal Utilities for NAT* (STUN)
    - *Interactive Connectivity Establisment* (ICE)
    - Discovery: Multicast oder ein Nameserver
- Fallback: Separater Kommunikationskanal zu Leitstelle
  - Zentrale Steuerung
  - Kann im Notfall eingreifen

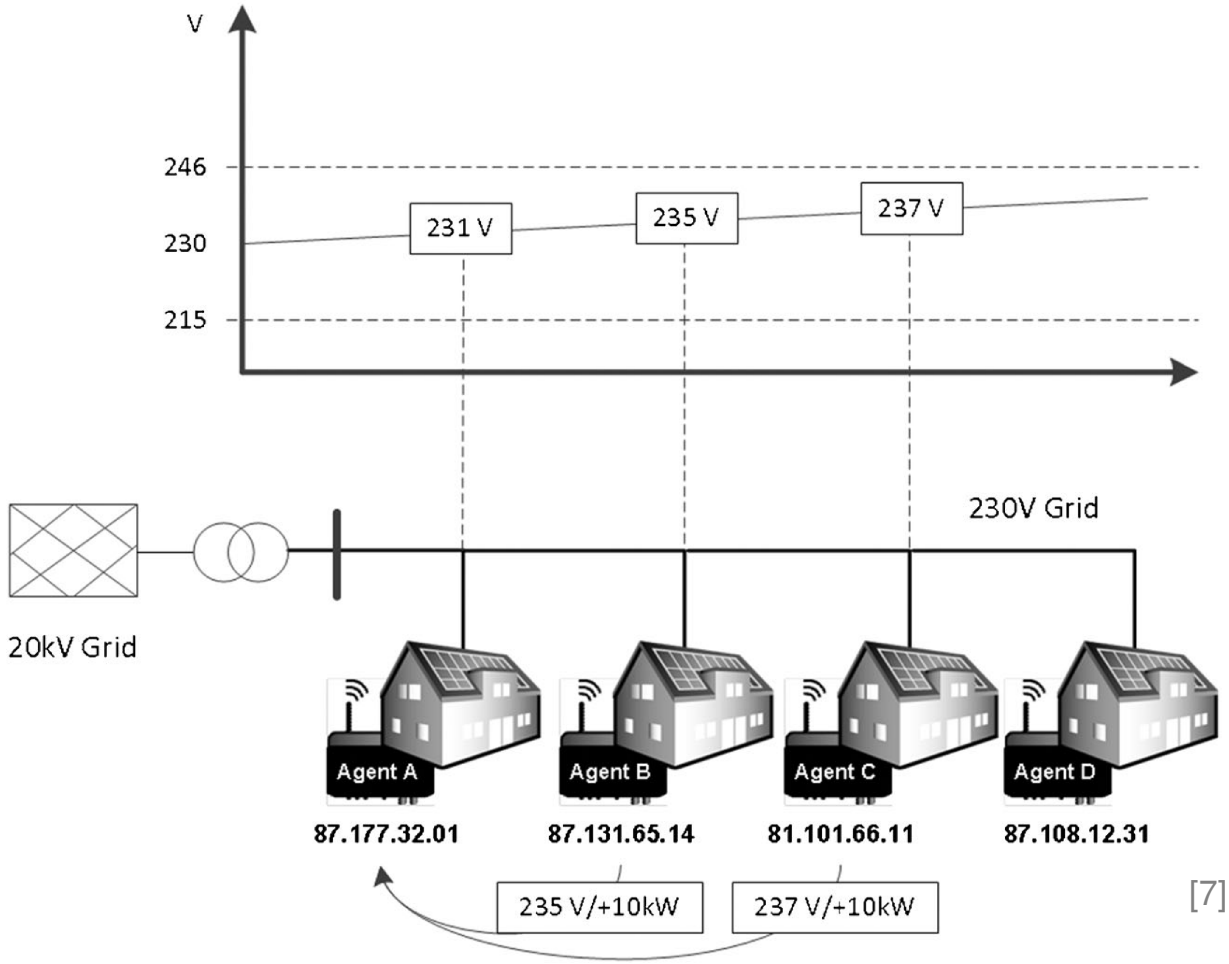
Agent-basierte Netze (2) – Kommunikation



[7]

Agent-basierte Netze (3) – Lokalisierung

- Agenten müssen Netzstruktur kennen
- Lokalisierung eines Agenten im Netz: Spannung mit anderen Agenten vergleichen



[7]

COHDA

Combinatorial **H**euristics for **D**istributed **A**gents

- Uni Oldenburg
- Verschiedene Paper
  - Approaching Decentralized Demand Side Management via Self-Organizing Agents [HVS11]
  - Evaluation of a Self-Organizing Heuristic for Interdependent Distributed Search Spaces [HSL13]
  - A Decentralized Heuristic for Multiple-Choice Combinatorial Optimization Problems [HLS14]
  - Hybrid Multi-ensemble Scheduling [BL17]

COHDA

Combinatorial **H**euristics for **D**istributed **A**gents

- Algorithmus zur selbstorganisierten Planung eines VPPs
- Vorgegebene Zielfunktion: Fahrplan
- Agent-basiert
  - Lokale Einschränkungen können berücksichtigt werden
  - Einschränkungen müssen nicht global bekannt sein $\rightarrow$ Privatsphäre
- Agenten versuchen mit kollektivem Verhalten globale Zielfunktion zu erfüllen

COHDA – Motivation aus der Natur

Beispiel Fischschwarm

- Schwarm soll
  - Hindernissen ausweichen
  - Teilung verkraften
- Schwarmbildung nach lokalen Regeln

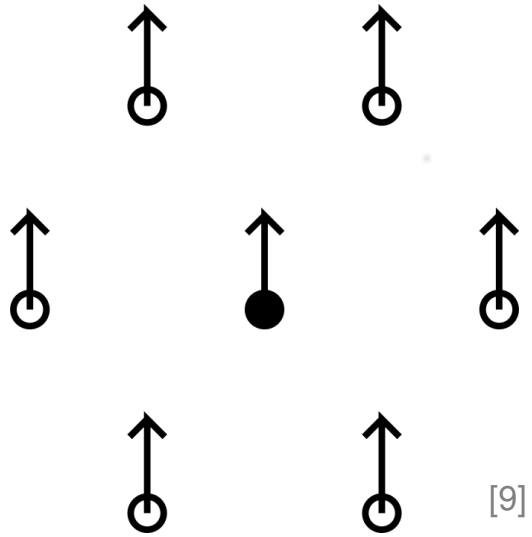
1. Kollisionsvermeidung
2. Geschwindigkeitsanpassung an Fische in Nachbarschaft
3. Schwarmzentrierung: möglichst nah an Fische in Nachbarschaft

COHDA – Motivation aus der Natur

Beispiel Fischschwarm

- Kein Fisch braucht Informationen über gesamtes System
- Aktionen einzelner Fische lösen Reaktionen der Fische in Nachbarschaft aus
- Hält sich jeder Fisch an die Regeln bildet sich ein funktionierender Schwarm

[8]



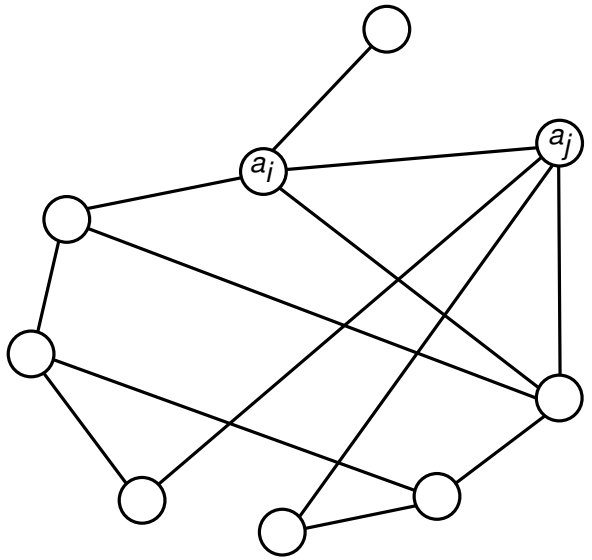
[9]

COHDA

Vorraussetzungen:

- Jeder Agent ist verantwortlich für eine Einheit (z.B. BHKW)
  - Einheit kann entweder Strom produzieren, verbrauchen oder beides
- Jeder Agent kann mit einer Menge von anderen Agenten kommunizieren $\rightarrow$ Nachbarschaft $\mathcal{N}_i$
- Kann durch ungerichteten Graph dargestellt werden:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}),$$
$$a_i \in \mathcal{V}, e = (a_i, a_j) \in \mathcal{E}$$
$$\mathcal{N}_i = \{a_j | (a_i, a_j) \in \mathcal{E}\}$$



COHDA

Ziel:

1. Jeder Agent a_i hat einen Fahrplan p_i ausgewählt, den seine Einheit U_i ausführen kann ohne technische Einschränkungen zu verletzen
2. Die Summe aller Fahrpläne ist möglichst nah an den Vorgaben des globalen Fahrplans ζ

- Ein Fahrplan hat z.B. 96 Zeitintervalle a 15 Minuten
- Jeder Agent a_i wählt genau einen Zeitplan p_i aus dem Suchraum $\mathcal{F}^{(U_i)}$ seiner Einheit U_i

$$\delta(\sum_{i=1}^{m} p_i, \zeta) \rightarrow min;$$

$$p_i \in \mathcal{F}^{U_i} \forall U_i \in U$$

COHDA

Karlsruher Institut für Technologie

Definitionen

- **Selektion:** Fahrplan p_i für Agent a_i
- **Status:** $\sigma_i =< p_i, \lambda_i >$ von a_i
  - Selektion p_i zum Zeitpunkt λ_i
- **Konfiguration:** $\Sigma = \{\sigma_i, \sigma_k, \ldots\}$
- **Lokale Konfiguration:** $\Sigma_i = \{\sigma_k | a_i \text{ kennt } a_k\}$
- **Beste Konfiguration:** $\Sigma_i^* = \{\sigma_i^*, \sigma_k^*, \ldots\}$
  - Beste Konfiguration, die ein Agent a_i gesehen hat

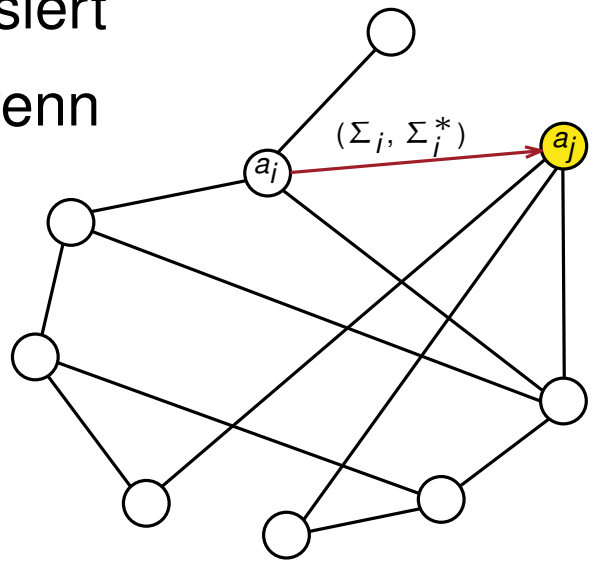
Drei Phasen: **(update)**, **(choose)**, **(publish)**

COHDA

Phase 1: **(update)**

- **Fall 1:** Agent a_j erhält Zielfunktion ζ
  - Startet Prozess, ζ wird lokal gespeichert
- **Fall 2:** Agent a_j erhält Informationen (Σ_i und Σ_i^*) von Nachbar a_i
  - Lokale Konfiguration (Σ_j) wird aktualisiert:
    - Status unbekannter Agenten wird hinzugefügt
    - Status bekannter Agenten wird aktualisiert
  - Beste Konfiguration (Σ_j^*) wird aktualisiert wenn
    - Σ_i^* besser als Σ_j^* oder
    - Σ_i^* hat mehr Elemente

Bei Änderung $\rightarrow$ Phase 3 **(publish)**

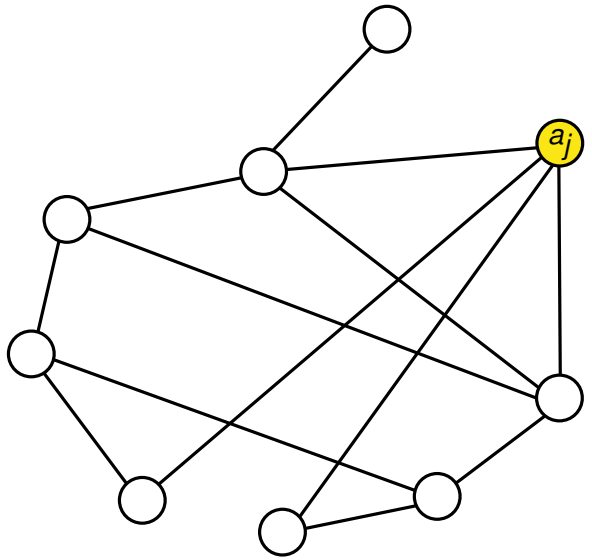


COHDA

Phase 2: **(choose)**

- Agent a_j versucht einen Fahrplan p_{neu} zu finden, der mit Σ_j den globalen Fahrplan ζ möglichst gut erfüllt
- **Fall 1:** Die neue Konfiguration ist besser als Σ_j^* und hat mindestens so viele Elemente
  - Σ_j wird mit Status $\sigma_j =< p_{neu}, \lambda_j + 1 >$ aktualisiert
  - $\Sigma_j^* = \Sigma_j$
- **Fall 2:** Die neue Konfiguration ist schlechter
  - Kehre zu Selektion p_j aus Σ_j^* zurück

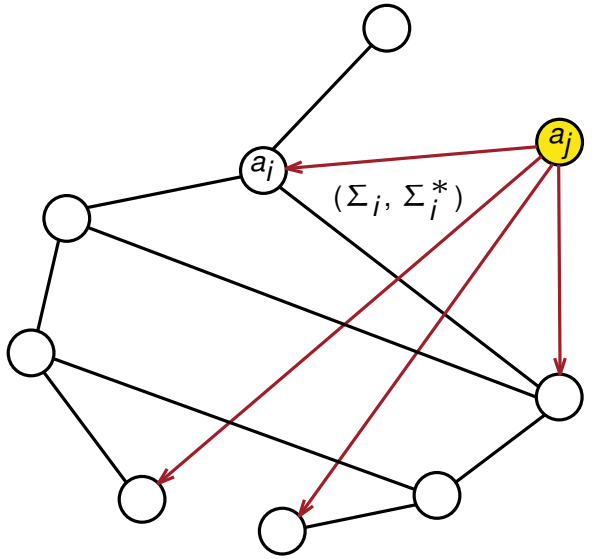
Bei Änderung $\rightarrow$ Phase 3 **(publish)**



COHDA

Phase 3: **(publish)**

- a_j schickt Σ_j^* und Σ_j an alle Nachbarn



COHDA

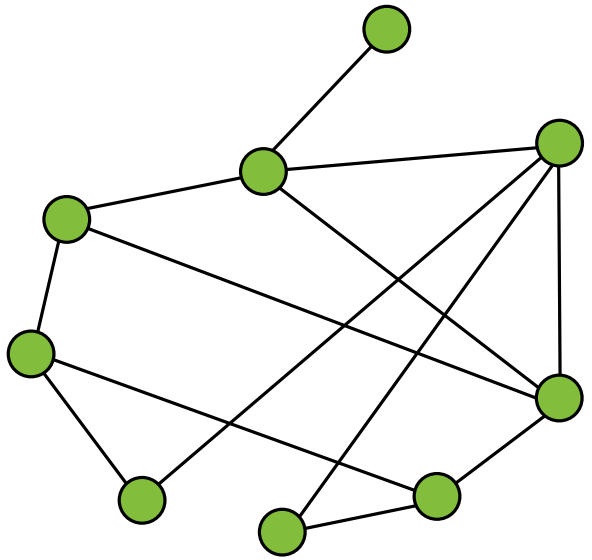
Phase 3: **(publish)**

- a_j schickt Σ_j^* und Σ_j an alle Nachbarn

Konvergenz:

- Nur bessere Konfigurationen überstehen Selektionsprozess
- Agenten einigen sich auf gemeinsame beste Konfiguration

Terminierung: Inaktivität aller Agenten

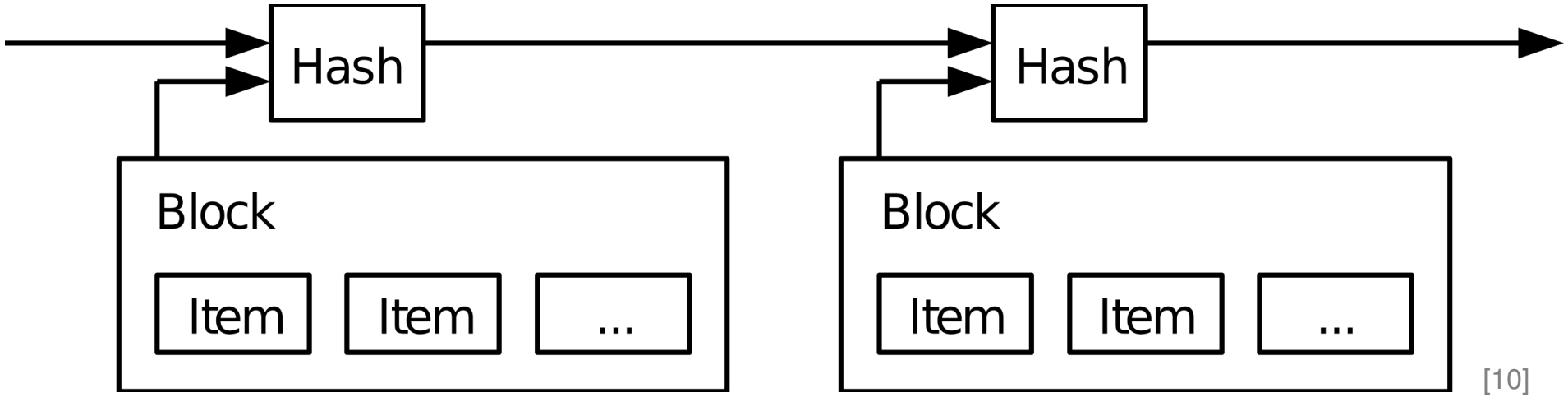


Übersicht

- Einführung: Begriffe
- Agent-basierte Netze
- COHDA: Dezentrale, selbstorganisierte Fahrplangenerierung
- Blockchain
  - Was ist das?
  - Beispiele: Bitcoin & Ethereum
  - Smart Contracts
  - ADMM mit Blockchain

Blockchain

- Gilt als ”Enabler”
- Dezentrale Buchführung
- Jeder Block referenziert vorherigen Block
- Änderung: alle nachfolgenden Blöcke müssten neu berechnet werden
- Konsens aller Parteien mittels kryptographischer Verfahren
- Bekannteste Anwendung: Bitcoin



[10]

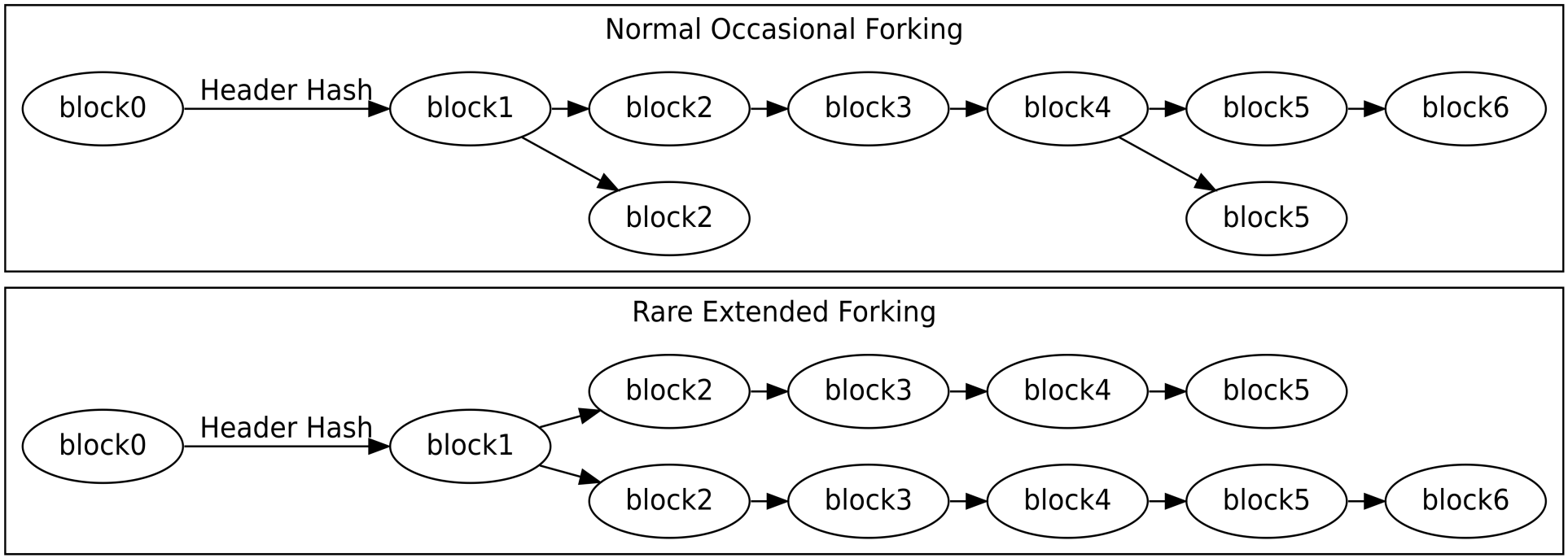
Blockchain – Bitcoin (BTC)

- Veröffentlichung 2009 von Satoshi Nakamoto [Nak08]
- Kryptowährung
  - Historie aller Transaktionen statt physischem Geld
  - Blockchain als öffentliches Transaktionsverzeichnis
    - **Transaktion:** Überweisung von Bitcoin von einer Adresse zu einer anderen
    - **Adresse:** Assoziiert mit Public-Key
    - Jede Transaktion ist mit Private-Key signiert
    - Neue Transaktionen werden als Block angefügt, mit Referenz auf vorherigen Block
  - Ein verteiltes, dezentrales Netzwerk an Rechnern sorgt für Verfügbarkeit und Integrität: **Bitcoin-Miner**

Blockchain – Proof of Work

1. Neue Transaktionen werden durch das Netz propagiert
2. Jeder Miner versucht einen Block zu generieren
   - Gesucht: Hash des Blockes in bestimmter Form
   - Enthalten: alle neuen Transaktionen, Hash des vorherigen Blocks, Nonce
   - ”Work”: so lange neu hashen bis ein passender Hash gefunden wurde
3. Valider Block wird durch das Netz propagiert. Belohnung: neue BTC
4. Konsens
   - Es kann auch mehrere valide Blöcke geben
   - Jeder Miner Entscheidet sich für einen validen Block
   - Mehrheit entscheidet
   - Es wird immer die längste Kette verwendet

Blockchain – Proof of Work



[11]

4. Konsens
   - Es kann auch mehrere valide Blöcke geben
   - Jeder Miner Entscheidet sich für einen validen Block
   - Mehrheit entscheidet
   - Es wird immer die längste Kette verwendet

Blockchain – Ethereum

- Blockchain nicht nur als Währung
- Eigene Währung ”Ether”
- Eigene Blockchain welche die dezentrale Ausführung beliebiger Programme erlaubt
- Smart Contracts

Blockchain – Smart Contracts

- Vertrag
  - Einigung von mehreren Personen
  - Notar bestätigt Zeit und Ort des Abschlusses
- Smart Contract: Software übernimmt Aufgabe von schriftlichem Vertrag
  - Damit: Maschine zu Maschine (M2M) Verträge
  - Automatische Überprüfung der Vertragsbedingungen

Blockchain – Ethereum Smart Contracts

- Ethereum Smart Contract: Autonome Einheit, welche bestimmte Dinge in Auftrag eines Nutzers erfüllen kann
  - Beliebiger Code kann ausgeführt werden
    - Muss deterministisch sein
    - Rechenzeit und Speicher muss mit Ether bezahlt werden
  - Aktivierung von Smart Contracts:
    - Vom Nutzer direkt
    - Von anderen Smart Contracts
    - Verbindung zur realen Welt: Orakel
  - Werte können in Blockchain abgelegt werden
  - Jeder Miner führt Code aus Smart Contract aus

ADMM mit Blockchain

- **A**lternating **D**irection **M**ethod of **M**ultipliers
  - Schwere konvexe Optimierungsprobleme werden in einfachere Unterprobleme Geteilt
  - Koordination durch Aggregationsschritt
  - Langsame Konvergenz
  - Garantiert globales Optimum
- Blockchains for decentralized optimization of energy resources in microgrid networks [MMM17]
  - ADMM zur Fahrplangenerierung und Abrechnung in Microgrid
  - Agent-basiert
  - Smart Contract in Blockchain übernimmt Aggregationsschritt
  - Smart Meter als Orakel
  - Lokale Variable x_i und globale Variable z

ADMM mit Blockchain – Algorithmus

repeat
- **begin** P_i**:**
  - Lokale Optimierung: Berechne x_i und sende an Smart Contract S_1
- **begin** S_1**:**
  - aktualisiere z
  - **if** *threshold* **then**
    - Berechne Fahrplan und Abrechnung
    - Sende Fahrplan an S_2

until *threshold*;

begin M_i**:** alle Smart Meter
- Zeichne Energieverbrauch auf
- Sende Verbrauch an Smart Contract S_2

begin S_2**:** Abrechnung in Blockchain
- Vergleiche Fahrplan mit Smart Meter Daten
- Berechne Bezahlungen und transferiere Geld

ADMM mit Blockchain: Warum?

- Zentrale Autoritäten handeln oft nicht in Bestem Interesse der Teilnehmer
- Alle Vorgänge sind transparent einsehbar
- Abrechnung direkt integriert
- Problem: nur day-ahead Planung möglich

Blockchain – Alternative Ansätze

- Blockchain im Energiesektor: Share&Charge
  - Verwaltung und Abrechnung von Ladestationen
  - Ethereum Smart Contracts
  - Ins Leben gerufen von RWE-Tochter Innogy

[12]

Problemdiskussion

- Stromverbrauch
  - Bitcoin verbraucht momentan mehr Strom als Irland
  - Alternative zu ”Proof of Work”: ”Proof of Stake”
- Anzahl Transaktionen
  - Bitcoin: 7/s
  - Ethereum: 20/s
  - Vergleich VISA: $>$10k/s
- Sicherheit
  - 51% Attacke
- Variante: private Blockchain

Zusammenfassung & Ausblick

- Stromnetz im Wandel: Hin zu verteilten, dezentralen Organisationsstrukturen
- Multi-Agenten-Systeme
- COHDA
  - Selbstorganisierte Fahrplangenerierung nach vorgegebener Zielfunktion
  - Lokale Einschränkungen müssen nicht global bekannt sein $\rightarrow$ Privatsphäre
- Blockchain als ”Enabler”
  - ADMM mit Smart Contracts
  - Transparenz
  - Abrechnung integriert
- Dezentralisierung vom Netzbetreiber oft nicht gewollt

Referenzen

Paper:
[HVS11] Approaching decentralized demand side management via self-organizing agents
[HSL13] Evaluation of a Self-organizing Heuristic for Interdependent Distributed Search Spaces
[HLS14] A decentralized heuristic for multiple-choice combinatorial optimization problems
[BL17] Hybrid Multi-ensemble Scheduling
[RBF14] Design of a Multiagent-Based Voltage Control System in Peer-to-Peer Networks for Smart Grids
[ASSR14] Robust scheduling in a self-organizing hierarchy of autonomous virtual power plants
[MMM17] Blockchains for decentralized optimization of energy resources in microgrid networks
[Nak08] Bitcoin: A peer-to-peer electronic cash system
Bilder:
[1] ABB: http://new.abb.com/grid/events/cigre-2016/microgrids Abgerufen am 08.01.2018
[2] ABB: http://new.abb.com/ch/smart-grids/smart-grid-technologien/intelligente-gebaeude/netz-der-zukunft Abgerufen am 08.01.2018
[3] Wikipedia: https://de.wikipedia.org/wiki/Client-Server-Modell Abgerufen am 08.01.2018
[4] Wikipedia: https://de.wikipedia.org/wiki/Peer-to-Peer Abgerufen am 08.01.2018
[5] [ASSR14]
[6] EBL: http://blog.ebl.ch/durch-regelenergie-zur-sicheren-stromversorgung/ Abgerufen am 08.01.2018
[7] [RBF14]
[8] Wikipedia: https://en.wikipedia.org/wiki/Shoaling_and_schooling Abgerufen am 08.01.2018
[9] [HVS11]
[10] [Nak08]
[11] https://bitcoin.org/en/developer-guide#block-height-and-forking Abgerufen am 08.01.2018
[12] slock.it:
https://blog.slock.it/share-charge-launches-its-app-on-boards-over-1-000-charging-stations-on-the-blockchain-ba8275390309
Abgerufen am 08.01.2018

Zusatz – ADMM

- ADMM: Allgemeine Form

$$\min_{x,y} f(x) + g(z)$$

$$Ax + Bz = c$$

$$L_\rho(x, z, \xi) := f(x) + g(z) + \xi^T(Ax + Bz - x) + \rho/2\|Ax + Bz - c\|^2$$

$$x^{k+1} = arg\min L_\rho(x, z^k, \xi^k)$$

$$z^{k+1} = arg\min L_\rho(x^{k+1}, z, \xi^k)$$

$$\xi^{k+1} = \xi^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$$