

# IRPsim – Eine webbasierte Ausführungsinfrastruktur für Energiesystemmodelle

*Stefan Kühne<sup>1,2</sup>, David Georg Reichelt<sup>1,2</sup>, Andreas Kluge<sup>1,2</sup>, Moritz Engelmann<sup>1,2</sup>*

## Highlights

- Zur nutzerfreundlichen Anwendung von Energiesystemmodellen wie IRPopt und IRPact, wurde die webbasierte Simulationsinfrastruktur IRPsim entwickelt.
- IRPsim verbirgt die Komplexität des zugrundeliegenden Formalismus durch nutzerfreundliche Ein- und Ausgabeelemente.
- IRPsim reduziert den notwendigen Konfigurationsaufwand durch Mechanismen zur Wiederverwendung von Konfigurationswissen.
- IRPsim ermöglicht über einen modellgetriebenen Softwareentwicklungsprozess die effiziente Umsetzung von Modellanpassungen und -erweiterungen.

## Einleitung

Energiesystemmodelle wie deeco,<sup>3</sup> DER-CAM,<sup>4</sup> IRPopt<sup>5</sup> oder IRPact<sup>6</sup> charakterisieren Energiesysteme aus verschiedenen Modellierungsperspek-

---

<sup>1</sup>Universitätsrechenzentrum Leipzig, Universität Leipzig

<sup>2</sup>Institut für Angewandte Informatik e. V.

<sup>3</sup>Vgl. Bruckner, 1997.

<sup>4</sup>Vgl. Stadler u. a., 2014.

<sup>5</sup>Vgl. Scheller, Burgenmeister u. a., 2018.

<sup>6</sup>Vgl. Johannig u. a., 2020.

tiven. Aspekte dieser Perspektiven sind:

- Marktteilnehmer, von denen in unterschiedlichen Rollen verschiedene Dienstleistungen entlang der Wertschöpfungskette abgeleitet werden (kommerzielle Akteure),
- technische Komponenten, aus denen relevante energietechnische Prozesse abgeleitet werden (Engineering-Komponenten),
- Energieflüsse und andere Komponentenbeziehungen innerhalb der Energiesystemkonfiguration (Komponentenbeziehungen),
- vertragliche Beziehungen zwischen Akteuren, die die jeweilige Leistungserbringung umfassen (operative Koordinationen),
- regulatorische sowie produktbezogene Anforderungen des Geschäftsumfelds (Marktprinzipien) und
- kontextabhängige Vorgaben räumlicher, ökologischer und zeitlicher Gegebenheiten (kommunale Kontexte).

Die Verwendung derartiger Energiesystemmodelle erfordert umfangreiche Eingabedaten und Konfigurationsinformationen über Akteure, Komponenten, Komponentenbeziehungen und Restriktionen des untersuchten Energiesystems. Darüber hinaus sind Annahmen und Prognosen über die zukünftige Entwicklung der Energiesysteme erforderlich, z. B. bezüglich der Entwicklung von Preisen, Steuern, regulatorischen Rahmenbedingungen und Märkten. Dies ist unter anderem bei Analysen zu innovativen Geschäftsmodellen wie Nachbarschaftsspeicher, Lastverschiebung, Eigenverbrauch, Direktnutzung und Direktvermarktung ersichtlich.<sup>7</sup> Der praktische Einsatz in kommunalen und industriellen Nutzungsszenarien bringt zudem Herausforderungen im Datenmanagement mit sich, darunter das Management von Stammdaten, die Organisation von Datenworkflows, die Integration von Daten aus unterschiedlichen Quellen und der Umgang mit der Datenqualität. Diese Aspekte stellen herausfordernde Aufgaben dar und müssen durch eine leistungsfähige Software-Infrastruktur unterstützt werden.

In diesem Beitrag stellen wir den generischen IRPsim-Ansatz zur Unterstützung der Ausführung von Energiesystemmodellen am Beispiel der Modelle IRPopt und IRPact vor. Zunächst konzentrieren wir uns auf Anforderungen des zugrundeliegenden Softwareentwicklungsprozesses

---

<sup>7</sup>Vgl. Scheller, Burkhardt u. a., 2020; Scheller, Krone u. a., 2018; Scheller, Reichelt u. a., 2017.

zur Abbildung der oben genannten Aspekte des Lebenszyklus. Anschließend beschreiben wir die entwickelten Konzepte zur Unterstützung der Anforderungen an Benutzerfreundlichkeit und Flexibilität.

## **Nicht-funktionale Anforderungen an eine webbasierte Simulationsinfrastruktur**

Wesentliche nicht-funktionale Anforderungen an eine webbasierte Simulationsinfrastruktur sind Benutzerfreundlichkeit und Flexibilität. Die bereitgestellten Benutzeroberflächen sollen sicherstellen, dass alle notwendigen Konfigurations- und Parametrierungsaufgaben mit möglichst geringem manuellem Aufwand durchgeführt werden können. Hierzu gehören Automatismen und intelligente Methoden zur Modellparametrisierung, domänenspezifische Abstraktionen für Modellparameter und die Validierung domänen-spezifischer Randbedingungen.

Gleichzeitig darf die Agilität der Analyseprozesse nicht eingeschränkt werden. Die Analyseaufgaben und -fragen für verschiedene Stakeholder entwickeln sich im Laufe der Zeit ständig weiter. Dies erfordert Änderungen an Eingabeschnittstellen, Modellparametern, Visualisierungen und sogar Änderungen an der Laufzeitumgebung, die durch Speicher- und Rechenressourcen gekennzeichnet ist. Die bereitgestellte Software-Infrastruktur muss daher die Flexibilität und Skalierbarkeit bieten, um solche Änderungsanforderungen abzubilden.

Das einer Simulationsinfrastruktur zugrunde liegende Energiesystemmodell bestimmt mit seinen Vorgaben hinsichtlich Ein- und Ausgangsparametern und Laufzeitumgebung (z. B. Solver) maßgeblich den Aufbau und das Design der Softwarekomponenten für den Umgang mit Ein- und Ausgabedaten, die Datenspeicherung, Visualisierung und Darstellung von Ergebnissen. Die Abhängigkeit der Softwareumgebung vom Energiesystemmodell impliziert, dass Änderungen am Modell direkte Auswirkungen auf eine oder mehrere Softwarekomponenten haben können. Beispielsweise erfordert das Hinzufügen eines Modelleingabeparameters neue Eingabefelder, ggf. neue Visualisierungsarten, Änderungen an Strukturübersichten und Aktualisierungen von Abhängigkeiten zu anderen Parametern. Tech-

nisch kann das Hinzufügen eines Eingabeparameters Änderungen an der Speicherinfrastruktur und den Kommunikationsschnittstellen erfordern.

Nach unseren Erfahrungen unterliegt ein Energiesystemmodell zudem kontinuierlichen Veränderungen in der praktischen Anwendung. Aus technischer Sicht impliziert dies die Notwendigkeit eines Life-Cycle-Managements der Software-Code-Infrastruktur, welche das Modell und die abhängigen Softwarekomponenten interoperabel und synchron hält.

Um dieses Problem anzugehen, folgen wir einem modellgetriebenen Softwareentwicklungsprozess.<sup>8</sup> Innerhalb dieses Ansatzes werden wesentliche Teile der zu entwickelnden Softwareinfrastruktur automatisch aus einer zugrunde liegenden domänenspezifischen Spezifikation generiert. Dies entspricht der sogenannten generativen Architektur, die das Konzept einer Abbildung zwischen einem Problemraum und einem Lösungsraum beschreibt. Der Problemraum stellt eine Reihe von domänenspezifischen Abstraktionen und Konzepten dar, mit denen Fachexperten ihr technisches Problem ausdrücken können. Der Lösungsraum hingegen besteht aus implementierungsorientierten Abstraktionen, die für eine bestimmte Laufzeitumgebung instanziiert werden können (z. B. basierend auf Datenbankkomponenten, Middleware-Infrastruktur, Frontend-Technologien). Abhängig von der konkreten Spezifikation des Problemraums wird über ein Mapping automatisch eine passende Lösung generiert. Das Mapping generiert schematische Codefragmente, die in eine generische Infrastruktur eingebettet sind und mit individuellem, handgeschriebenem Softwarecode ergänzt werden können.<sup>9</sup>

## Szenario-basierte Konfiguration

Mit Hunderten von Eingangsparametern – Skalare, Arrays, Sets, Zeitreihen – erfordert die Anwendung eines Energiesystemmodells in der Praxis einen erheblichen Konfigurationsaufwand, um reale Geschäftsszenarien zu bewerten. Ein wichtiger Aspekt bei der Reduzierung des Konfigurationsaufwands ist die Wiederverwendung von Konfigurationswissen. Um diese

---

<sup>8</sup>Vgl. Czarnecki, 2004.

<sup>9</sup>Vgl. Kühne u. a., 2019.

Anforderung hinsichtlich Nutzerfreundlichkeit zu erfüllen, verwenden wir den Ansatz der szenariobasierten Konfiguration.

Die Idee dahinter ist, dass die zahlreichen Parameter eines Modells nicht unabhängig voneinander in eine Spezifikation einer Simulation einfließen. Vielmehr bestimmen grundsätzliche Annahmen über zukünftige Entwicklungen – z. B. Annahmen über die Entwicklung von Preisen, Technologien oder Märkten – die konkrete Zuordnung von Parametern. Typische Annahmen eines Geschäftsszenarios können zusammengefasst und beispielsweise als „business as usual“, „collaborative economy“ oder „consumer efficiency“ deklariert werden. Für jedes Geschäftsszenario können vordefinierte Datenwerte (Skalare, Zeitreihen) mit Eingabeparametern eines Modellparametersatzes für bestimmte Prognosehorizonte verknüpft werden. Die Anwendung eines Geschäftsszenarios in der Parametrisierungsphase des Modells führt zu einer Zuordnung der vordefinierten Werte, die dann vom Benutzer verfeinert und angepasst werden können.

Da die Datenwerte dynamisch aus einer Datenbank geladen werden, nennen wir dieses Verhalten „Pull-Funktionalität“. Die Herausforderung dieses Ansatzes liegt im Datenmanagement, insbesondere im Stammdatenmanagement, das die konsistente und vollständige Zuordnung der Werte gewährleisten muss und das wir im Folgenden beschreiben.

## Datenmanagement

Die Grundlagen unseres Datenmanagementansatzes sind in folgendem Klassendiagramm (Abbildung 8.1) dargestellt. Das Kernkonzept ist Stammdaten (MasterData). Stammdateneinträge stellen Datenquellen dar, die Modelleingabeparametern zugeordnet werden können. Sie werden durch den Tupelnamen, den Typ (ein Verweis auf den Parameternamen des Modells) und das Referenzjahr (Verweis auf das Jahr, auf das sich die Prognose bezieht) eindeutig identifiziert. Für jede Stammdatenerfassung kann ein Datenmanager den Prognosezeitraum definieren, d. h. für wie viele Folgejahre Datensätze definiert werden können, und das Zeitintervall, in dem die Daten gespeichert werden. Optional können für Stammdateneinträge Set-Abhängigkeiten definiert werden. Diese Abhängigkeiten legen fest, für welche Menge und für welche Elemente der Menge die jeweilige Stammdateneingabe gelten soll. Diese Informationen werden in der

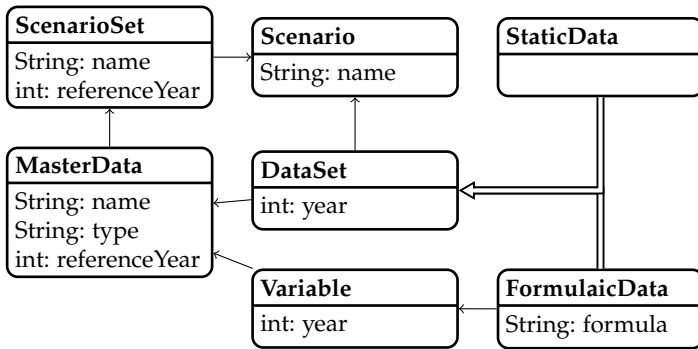


Abbildung 8.1.: Grundkonzepte des Datenmanagements in IRPsim. Eigene Darstellung.

IRPsim-Simulationsinfrastruktur zur Vorfilterung von Datensätzen und zur automatisierten Zuordnung innerhalb der Pull-Funktionalität verwendet.

Stammdateneinträge werden hierarchisch über die Relation „verfeinert“ organisiert. Die resultierende Hierarchie wird als Vererbungshierarchie interpretiert und sowohl auf Daten als auch auf Metadaten von Stammdateneinträgen angewendet. Beispielsweise werden Metadaten, die in einer bestimmten Stammdatenerfassung fehlen, aus der allgemeineren Stammdatenerfassung entnommen. Gleiches gilt für zugeordnete Daten. Für spezifischere Stammdateneinträge steht ein für eine allgemeine Stammdatenerfassung hinterlegter Datensatz zur Verfügung und kann dort verfeinert werden.

Das Konzept DataSet stellt Datenwerte (Skalare oder Zeitreihen) dar, die Modelleingabeparametern in Abhängigkeit vom referenzierten Szenario, den festgelegten Abhängigkeiten und dem Zeitraum zugewiesen werden. Datensätze können auf zwei verschiedene Arten ausgedrückt werden. StaticData-Definitionen stellen numerische Datenwerte dar, die statisch in eine Datenbank importiert werden. Eine FormulaicData-Definition ermöglicht einen algebraischen Ausdruck von Datenwerten. Dazu können vorhandene DataSets oder vordefinierte Konstanten, z. B. ein Zeitvektor, der einen Zwölf-Stunden-Zyklus darstellt, verwendet werden.

Mit Hilfe der algebraischen Darstellung können Abhängigkeiten zwischen

verschiedenen Skalaren oder Zeitreihen leicht erklärt und Datenwerte als Funktion einer beabsichtigten zeitlichen Entwicklung propagiert werden. Des Weiteren werden Datenbankredundanzen deutlich reduziert und die Flexibilität für den Anwender, bestehende Datendefinitionen zu manipulieren, erhöht. Die Auswertung der algebraischen Darstellungen erfolgt bei der Parameterübergabe an den Solver.

Abbildung 8.2 zeigt die Verwaltungsschnittstelle für Stammdateneinträge. Sie bietet Übersichten über definierte Metadateneinträge und gespeicherte Daten. In dieser Schnittstelle können FormulaicData-Einträge mithilfe eines Formeleditors ausgedrückt werden. Statische Datenelemente können über Excel-Tabellen importiert und exportiert werden. Für jede Metadateneingabe werden qualitätsrelevante Eigenschaften, z. B. Vollständigkeit, abgeleitet und eine zuständige Geschäftseinheit und Ansprechpartner zugeordnet. So können verschiedene Geschäftseinheiten an der Problemlösung teilnehmen, indem sie relevante Daten bereitstellen, um die Datenerfassung, -organisation und -weitergabe zu erleichtern.

Name	Parametertyp	Bezugsjah...	Prognoseh...	Auflösung	Vollständigke...	Szenarien	Set 1	Abstrak...
	par_SOC_DES_I X							
✓ Tesla Powerwall	par_SOC_DES_ES...	2015	0	Jahr	66.67%	Business as Usual, Collabo...	set_tech_DES...	false
✓ Tesla Powerwall	par_SOC_DES_ES...	2015	0	Jahr	66.67%	Business as Usual, Collabo...	set_tech_DES...	false
✓ Tesla Powerwall	par_SOC_DES_ES...	2015	0	Jahr	66.67%	Business as Usual, Collabo...	set_tech_DES...	false
✓ Quartierspeicher (I)	par_SOC_DES_ES...	2015	0	Jahr	33.33%	Business as Usual, Collabo...	set_tech_DES...	false
✓ Quartierspeicher (I)	par_SOC_DES_ES...	2015	0	Jahr	33.33%	Business as Usual, Collabo...	set_tech_DES...	false
✓ Quartierspeicher (I)	par_SOC_DES_ES...	2015	0	Jahr	33.33%	Business as Usual, Collabo...	set_tech_DES...	false

Abbildung 8.2.: Verwaltungsschnittstelle für Stammdateneinträge. Eigene Darstellung.

## Graphische Repräsentation von Ein- und Ausgabedaten

Die graphische Repräsentation wird durch drei Werkzeuge unterstützt: Die Visualisierung des IRPopt-Systemmodells, die Visualisierung des IRPact-Agentengraphen und die Datenreihenvisualisierung.

**IRPopt-Systemmodell** Der sogenannte Energiesystemmodellgraph bietet eine intuitive Möglichkeit, gewünschte Systemkonfigurationen zu überprüfen und überlappende Beziehungen von Parametereinstellungen zu validieren (Beispiel siehe Abbildung 8.3). Um das Energiesystemmodell zu definie-

ren, muss für jedes Energiesystemelement, z. B. Umwandlungstechnologien oder Bedarfslasten, definiert werden, ob es existiert und wie es mit anderen Elementen verbunden ist. Die allgemeine Verwendung eines Energiesystemelements wird durch den Parameter *par\_X\_pss\_model* definiert und die Verknüpfungen werden durch eine Reihe von Parametern definiert, die mit *par\_X\_* beginnen, z. B. definiert *par\_X\_E\_DES\_fromES\_NS\_toEGrid\_energyLink*, ob eine Energieverbindung vom Energiespeicher (ES) zu einem Prosumstorer (NS) besteht. Beide werden durch boolesche Elemente definiert, die durch 0 oder 1 dargestellt werden. Da die Einstellung aller Parameter zeitaufwändig und fehleranfällig ist, unterstützt der Energiesystemmodellgraph das Auffinden von Problemen durch Visualisierung des konfigurierten Energiesystemmodells. Um das Verständnis des Energiesystems zu erleichtern, enthält die Grafik folgende Erweiterungen:

- **Energieflüsse:** Energieflüsse werden als Kanten des Graphen visualisiert. Die Farbe definiert die Art des Energieflusses, z. B. können graue Kanten Stromflüsse und blaue Ränder Wärmeströme darstellen.
- **Komponententyp:** Der Komponententyp wird durch die Form der Knoten visualisiert, z. B. kann eine dreieckige Pyramide ein Nettoelement darstellen und ein Kreis mit Brüchen kann Marktelemente darstellen.
- **Akteurssouveränität:** Die Akteurssouveränität wird durch die Farbe eines Knotens dargestellt, z. B. kann Blau die Vertriebs- und Marktseite darstellen und grün die Kundengruppe p1.
- **Technologien:** Die Technologien werden zusätzlich als Icons dargestellt, die z. B. Energienetz, Energiemarkt oder Wärmemarkt darstellen.

Insgesamt können Anwender mit Hilfe des Energiesystemmodellgraphen die Energie- und Finanzflüsse zwischen Marktsegmenten, Technologiekomponenten und Kundengruppen einer bestimmten Parameterkonfiguration visualisieren und auf Vollständigkeit überprüfen.

**IRPact-Agentenetzwerkgraph** Äquivalent zum Energiesystemmodellgraph in IRPopt ist es für den Nutzer von IRPact hilfreich, das Agentennetzwerk als Graph zu betrachten. Im Vergleich zum Systemmodell kann das Agentennetzwerk bedeutend größer werden, weshalb eine alternative



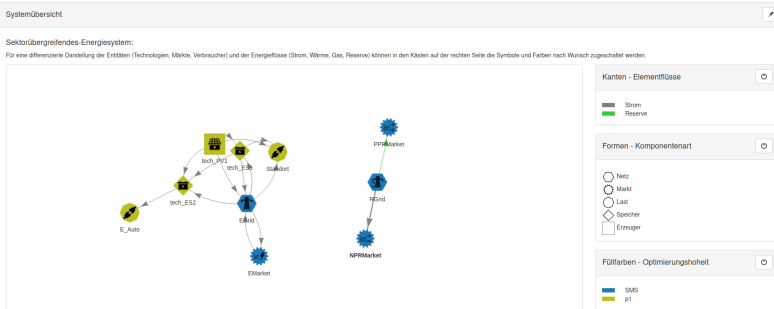


Abbildung 8.3.: Exemplarischer Energiesystemmodellgraph. Eigene Darstellung.

Möglichkeit der Darstellung benötigt wurde. Dafür wurde im Frontend die Möglichkeit geschaffen, ein durch das Backend generiertes Bild des Graphen anzuzeigen. Der Agentengraph wird dabei von IRPact selbst generiert und ist damit generisch; ein anderes Modell könnte folglich einen anderen Graphen generieren. Die Ansicht im Frontend verfügt über eine Zoom-Funktion, mit der die Größe des Graphenbilds geändert werden kann. Ebenfalls ist es möglich, das Bild lokal zu speichern. Die Abbildung 8.4 zeigt ein Beispiel für einen von IRPact visualisierten Graph, welcher das soziale Netzwerk der Agenten repräsentiert. Die Milieus sind dabei farblich kodiert und Kanten beschreiben potentielle Kommunikationspartner.

**Datenreihenvisualisierung** Zum Verständnis von Eingabe- und Ausgabedaten können Zeitreihen in einer zoombaren 2D-Visualisierung mit Aggregationsfunktionen wie Summe, Durchschnitt und Perzentilen oder in einer interaktiven 3D-Darstellung dargestellt werden. Die 3D-Anzeige ermöglicht die Visualisierung von Werten (auf z-Achse), die für spezielle Zeitintervalle aggregiert sind, z. B. um das Wochenintervall auf der y-Achse und die Stunde in der Woche auf der x-Achse zu visualisieren. Dies erleichtert die visuelle Identifizierung von Mustern, z. B. hoher Energieverbrauch zu Beginn jeder Woche oder jeden Tages. (siehe Abbildung 8.5).

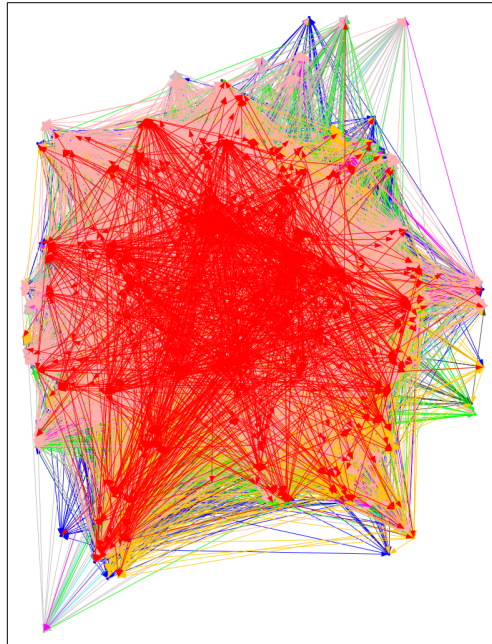


Abbildung 8.4.: Beispiel-IRPact-Agentennetzwerkgraph. Eigene Darstellung.

## Skalierbare Ausführungsumgebung

Skalierbarkeitsanforderungen für die IRPsim-Softwareumgebung sind (1) horizontale Skalierbarkeit (Hinzufügen weiterer Rechenknoten) entsprechend der Modellgröße und Anzahl der Jobs und (2) vertikale Skalierbarkeit (Skalierung durch Hinzufügen von mehr Leistung in Bezug auf CPU & RAM zu bestehenden Rechenknoten), z. B. zum Starten verschiedener Instanzen des IRPopt-Systems für unterschiedliche organisatorische Anforderungen. Die IRPsim-Softwarekomponenten wurden als Micro-Service-Architektur mit Docker und Docker Compose<sup>10</sup> implementiert. Die resultierende Software besteht aus drei zentralen Diensten: (1) dem UI-Service,

<sup>10</sup>Siehe <https://www.docker.com/>.

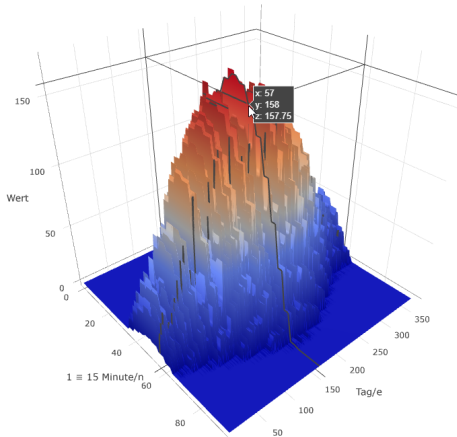


Abbildung 8.5.: 3D-Darstellung von Zeitreihen. Eigene Darstellung.

(2) dem Datenintegrationsdienst und (3) dem Datenverwaltungsdienst. Der UI-Dienst (1) basiert auf NodeJS<sup>11</sup> und stellt eine webbasierte grafische Benutzeroberfläche zur Verfügung. Der Backend-Service (2) – implementiert mit Jersey<sup>12</sup> – speichert Datenworkflows und verwaltet Simulationsläufe. Das Job-Management-Modul startet in einem anderen Prozess zusammen mit Job-Parametern. Daher sind Simulationsläufe von der Softwareumgebung isoliert und können auf verschiedenen Rechenknoten ausgeführt werden. Der Datenbankdienst (3) basiert auf dem Datenbanksystem MariaDB<sup>13</sup>.

## Danksagung

Dieser Beitrag wurde finanziert durch das Projekt „Smart Utilities and Sustainable Infrastructure Change“ (Antragsnummer 100378087 (SAB)).

---

<sup>11</sup>Siehe <https://nodejs.org/de/about/>.

<sup>12</sup>Siehe <https://projects.eclipse.org/projects/ee4j.jersey/>.

<sup>13</sup>Siehe <https://mariadb.org/>.

Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des vom Sächsischen Landtag beschlossenen Haushaltes.



DOI: <https://doi.org/10.30819/5413.08>

## Literatur

- Bruckner, T. J. C. (1997). »Dynamische Energie- und Emissionsoptimierung regionaler Energiesysteme«. Diss. Uni Würzburg.
- Czarnecki, K. (2004). »Overview of Generative Software Development«. In: UPP'04. Le Mont Saint Michel, France: Springer-Verlag, S. 326–341. ISBN: 3540278842. DOI: 10.1007/11527800\_25. URL: [https://doi.org/10.1007/11527800\\_25](https://doi.org/10.1007/11527800_25).
- Johanning, S., F. Scheller, D. Abitz, C. Wehner und T. Bruckner (2020). »A modular multi-agent framework for innovation diffusion in changing business environments: conceptualization, formalization and implementation«. In: *Complex Adaptive Systems Modeling* 8.1. DOI: 10.1186/s40294-020-00074-6.
- Kühne, S., F. Scheller, H. Kondziella, D. G. Reichelt und T. Bruckner (2019). »Decision support system for municipal energy utilities: Approach, architecture, and implementation«. In: *Chemical Engineering & Technology* 42,9, S. 1914–1922. DOI: 10.1002/ceat.201800665.
- Scheller, F., B. Burgenmeister, H. Kondziella, S. Kühne, D. G. Reichelt und T. Bruckner (2018). »Towards integrated multi-modal municipal energy systems: An actor-oriented optimization approach«. In: *Applied Energy* 228, S. 2009–2023. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2018.07.027. URL: <https://www.sciencedirect.com/science/article/pii/S0306261918310559>.
- Scheller, F., R. Burkhardt, R. Schwarzeit, R. McKenna und T. Bruckner (2020). »Competition between simultaneous demand-side flexibility options: the case of community electricity storage systems«. In: *Applied Energy* 269, S. 114969. DOI: 10.1016/j.apenergy.2020.114969.
- Scheller, F., J. Krone, S. Kühne und T. Bruckner (2018). »Provoking residential demand response through variable electricity tariffs—a model-based assessment for municipal energy utilities«. In: *Technology and Economics of Smart Grids and Sustainable Energy* 3.1, S. 1–20.
- Scheller, F., D. G. Reichelt, S. Dienst, S. Johanning, S. Reichardt und T. Bruckner (2017). »Effects of implementing decentralized business models at a neighborhood energy system level: A model based cross-sectoral analysis«. In: *2017 14th International Conference on the European Energy Market (EEM)*. IEEE, S. 1–6. DOI: 10.1109/EEM.2017.7981910.
- Stadler, M., M. Groissböck, G. Cardoso und C. Marnay (2014). »Optimizing Distributed Energy Resources and building retrofits with the strategic DER-CAModel«. In: *Applied Energy* 132, S. 557–567. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2014.07.041. URL: <https://www.sciencedirect.com/science/article/pii/S0306261914007235>.