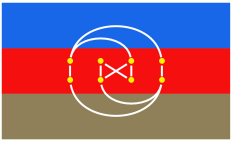
agents.oic.os



OiC.OS — Das Betriebssystem für Ökonomische Organisationen

Eduard Bretthauer · Viktor Winschel

Juni 2026

https://oicos.systems

I. → **Vorgehen:** unser Angebot, unsere universelle Lösung, Ihr Nutzen.

II. → **Technologie:** Agenten, die entscheiden, lernen und sich organisieren.

III. → **Aktuelles Projekt:** LiquiPool — der einfache Einstieg in das System.

Keine Theorie -
sondern ein laufendes System,
das die Mitarbeiter selbst programmieren.

Das Prinzip

1. Das ist unser Angebot

Wir liefern eine visuelle Programmiersprache, mit der sich Organisationen aus Agenten, wie nach dem Lego Prinzip, zusammensetzen lassen. Das Ergebniss IST das ERP System selbst.

- **Software ist kompositional (Modulsysteme vorhanden), Management bisher nicht.** Die Softwareindustrie hat Sprachen, Compiler und modulare Bausteine — in ökonomischer Steuerung und Rechnungswesen fehlt genau das.
- Wir bringen dieses Prinzip — *Kompositionalität* — erstmals in die Ökonomik, Managementsysteme, Buchhaltung und damit in die Ökonomie: komplexe Organisationen ergeben sich aus kleinen, wiederverwendbaren Teilen.
- Aus der Theorie bzw. Spezifikation ergibt sich das Modell also die Software, **automatisch, algorithmisch** — der Übersetzungsschritt geschieht per Compiler, er berechnet das Programm. Nicht in monatelangen Customisationschritten und Feedbackschleifen zwischen Management ↔ Customizer ↔ Programmierer.

2. Das ist unsere universelle Lösung

Ein Prinzip, für jede Größenordnung.

Prozesse in Unternehmen mit Kostenstellen, die Abteilung, das ganze Unternehmen, die Holding, die Lieferkette oder ganze Währungsräume werden aus wiederverwendbaren Agenten-Bausteinen zusammengesetzt — nur die Größenordnung ändert sich, aber nicht das Prinzip.

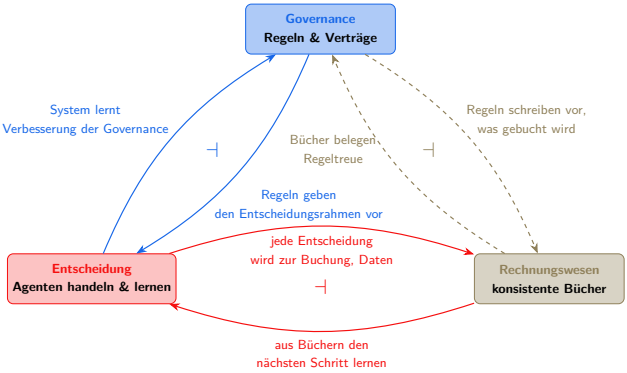
Drei Blickwinkel auf das System:

- **Governance** — Regeln und Verträge
- **Entscheidung** — Agenten entscheiden und lernen
- **Rechnungswesen** — Konsistente Bücher, Daten

Was das ermöglicht:

- Lernen: Entscheidung $\leftrightarrow$ Daten
- System lernt: Governance $\leftrightarrow$ Entscheidung
- Restrukturierung
- Skalierung

Konsistenz der drei Blickwinkel



Je zwei Bereiche sind **fest gekoppelt** (Symbol ⊣): Was der eine vorgibt, spiegelt der andere *als Konsistenzbedingung in beide Richtungen* zurück. **Genau deshalb bleiben Regeln, Entscheidungen und Bücher immer im Einklang** — Abweichungen werden gemessen, nicht erst im Nachhinein entdeckt.

Compliant by Decision nicht nur per ex post Audit.

3. Wie löst das Ihre Probleme?

Prozesse und die Restrukturierung selbst lassen sich wegen der Kompostionalität in Szenarien vor der Umsetzung simulieren. Das Ergebnis ist ein mitlaufender, geprüfter und mitwachsender Digital Twin Ihrer Organisation.

- **Szenarien rechnen:** Make-or-Buy, In-/Outsourcing, M&A, Lieferkette, Holdingstruktur — mit den internen und externen Buchungsfolgen jeder Variante.
- **Profitabilität verstehen:** wo der Profit entsteht, Verrechnungspreise, Deckungsbeiträge, Break-even.
- **Kein starres PowerPoint, kein statisches Excel im Laufwerk:** sondern ein digitaler Zwilling, der mitläuft, lernt und sich reorganisieren lässt.

Technologie

Welche Agenten gibt es?

Zwei Arten von künstlicher Intelligenz — wir ergänzen den aktuellen Hype:

A. Statistische KI

- Sprachmodelle (LLM)
- stark im Erkennen, Formulieren, Vorschlagen
- aber: kein Ziel, keine Konsistenz, keine Bücher

B. Logische (algebraische) KI — OiC.OS Agenten

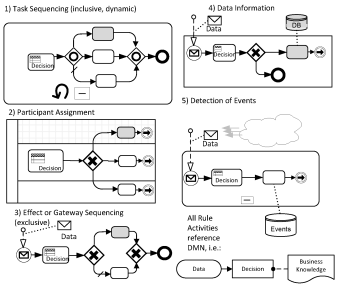
- ökonomisch spezialisierte **Agenten**
- lassen sich zusammensetzen (kompositional)
- können Rechnungswesen, Kostenrechnung und Finanzbuchhaltung

Das LLM redet —
OiC.OS Agenten rechnen, buchen und organisieren.
Stellen Wissen dar.

Beispiel: Wo heutige Prozess-Standards aufhören

BPMN (Business Process Model Notation)

DMN (Decision Model Notation)



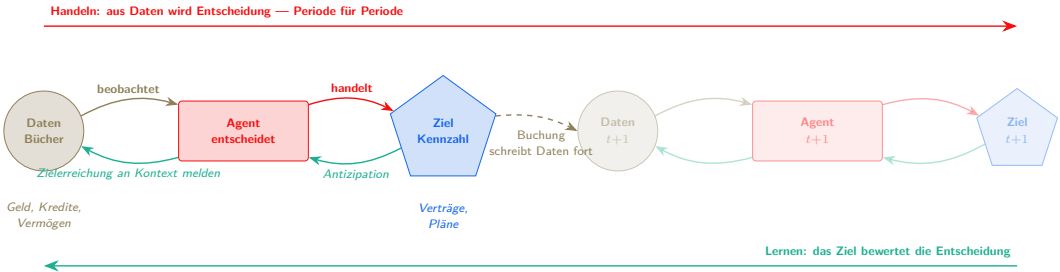
Was diesem Standard fehlt — und was OiC.OS kann:

- **Ziele** — warum wird wie entschieden?
- **Kompositionalität** — Bausteine zusammensetzen
- **Hierarchie** — Ebenen zusammensetzen
- **Verträge** — Regeln, Smarte (algorithmische) Kontrakte
- **Lernen** — Agenten verbessern ihre Entscheidungen
- **Einheiten, reale und Geld** — Rechnungseinheiten (Kostenstellen)
- **Konsistente Bücher** — Konsistenz über mehrere doppelte Buchführungssysteme (z.B. bei Schuldverträge, aka Quadruple Accounting)
- **Beweisbarkeit** — OiC.OS Software ist **beweisbar**, gegeben (die visuelle) Spezifikation, richtig implementiert

Was ermöglichen die explizit formulierten Ziele der OiC.OS Agenten?

- **Lernen wird erst durch Ziele möglich:** gelernt wird aus der Abweichung zwischen Ziel und tatsächlicher Realisierung.
- **Automatisierte Anpassung:** Ändern sich Teilziele in einem großen Netzwerkplan, passen sich die Entscheidungsknoten automatisch an — in BPMN/DMN erfordert das manuelle Eingriffe von Beratern oder Mitarbeitern.
- **Das Gesamtsystem lernt:** erst zusammen mit unserer kompositionalen Technologie führt dieses Lernen dazu, dass nicht nur einzelne Bausteine, sondern die Organisation als Ganzes lernt.

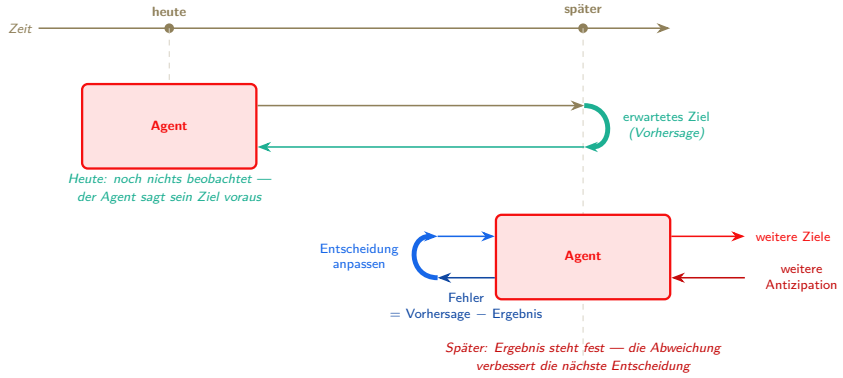
Was ist ein OiC.OS Agent? — Die zeitliche Komposition



Ein Agent beobachtet die Daten in seinem Kontext, **entscheidet** mit Hinblick auf sein **Ziel** und spiegelt, wie gut die Entscheidung zum Ziel beigetragen hat — das Rückwärtssignal macht Agenten kompositional und Organisationen lernfähig.

Zeitliche Komposition: Die **Buchung** schreibt die Bücher fort — das Ergebnis einer Periode wird zum Startzustand der nächsten. So reiht sich **derselbe Agent über die Zeit** zu einer durchgehenden Kette.

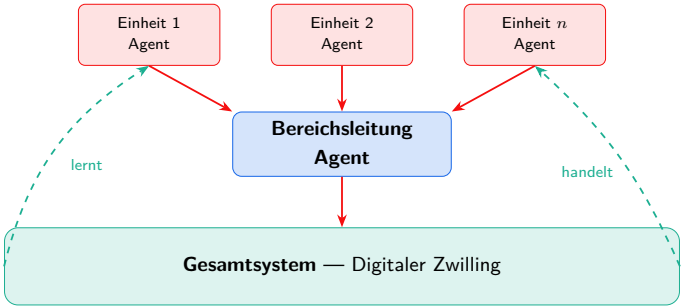
Wie ein Agent lernt: Erwartung trifft Ergebnis



Heute. Der Agent legt sich auf ein **erwartetes Ziel** fest — seine Vorhersage, bevor das Ergebnis bekannt ist.

Später. Das **tatsächliche Ziel** liegt vor. Die **Abweichung** (Ergebnis minus Vorhersage) fließt zurück und **passt die nächste Entscheidung an** — so lernt der Agent, Periode für Periode.

Was heißt kompositional? Warum Ziele?



- **Kompositional**: komplexe Prozesse in wiederverwendbare Bausteine zerlegen — und wieder zusammensetzen.
- Mehrere Agenten werden zu einem weiteren Agenten und System auf der nächsten Ebene.
- **Warum Ziele?** Ohne Zieldefinition kein Lernen. Ohne hierarchische Kompositionalität kein Systemlernen.
- Universalität: Kostenstelle → Abteilung → Unternehmen → Holding → Lieferkette → Währungsraum.

Wir wenden die Architekturmethode der Informatik auf die Ökonomik an.

Jeder Begriff aus Ihrer Organisation hat eine exakte Entsprechung in der Software-Architektur — dasselbe bewährte Bauprinzip, nur auf die Ökonomie angewandt:

Ökonomik	Informatik
Organisation — Regeln, Entscheidung, Bücher	Laufzeitumgebung, Logik im *Topos* mit *interner Typentheorie*
Konsolidierung: lokale Bücher zur Konzernbilanz	Verteilte Daten zu *einem* konsistenten Zustand
Planen vorwärts, aus Ist-Zahlen rückwärts lernen	Verlustfreies Hin- und Zurück *(Compiler ↔ Decompiler)*
Agent: handeln *und* bewerten am Ziel	Bidirektionales Programm — gleiche Form wie Backpropagation
Verträge: Rechte und Pflichten	Maschinell geprüfte Zusicherungen *(Propositions-as-Types)*
Eine Periode: lesen → entscheiden → buchen	Unfold-then-Fold ohne Zwischenspeicher *(Hylomorphismus)*
Gleiche Funktion auf jeder Ebene	Parametrischer Baustein *(polymorphe Funktion)*
Geschlossener Geldkreislauf	Rückkopplung als Fixpunkt *(Trace / fix)*

In Software seit Jahrzehnten Standard, nun in OiC.OS erstmals in der Steuerung von Organisationen — Spezifikation und Software werden eins, die Übersetzung erledigt der Compiler.

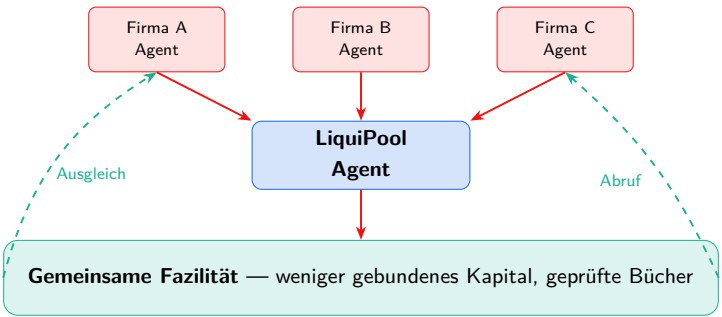
Aktuelles Projekt

Eine einfache Anwendung, um das komplexe System zu entwickeln.

Bisher nur in einem akdemischen Modell der Geldtheorie (5 Agenten, Sektoren, Banken, Zentralbank).

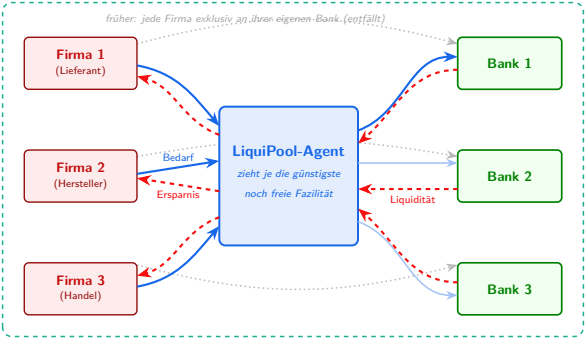
- **Einfacher Einstieg:** mehrere Firmen teilen sich eine gemeinsame Liquiditätsfazilität statt drei getrennter.
- **Das komplexe System entwickeln:** dieselbe Mechanik trägt später Kostenrechnung, Holding-Steuerung bis hin zur Bank-Logik.
- **Von Fachexperten nutzbar:** bedienbar ohne Programmierung (Low-Code / visuelle Modellierung).
- **Schulung am einfachen Beispiel:** Mitarbeiter lernen das System an einem überschaubaren, sofort nützlichen Fall.

LiquiPool: drei Firmen, ein Pool — ein Agent



Jede Firma meldet ihren **Liquiditätsbedarf**; der LiquiPool-Agent verteilt nach Vorrang und gleicht aus. Drei einzelne Fazilitäten kosten mehr als eine gemeinsame — **die Zusammensetzung schafft den Wert** ($3 \times 1 \neq 1 \times 3$). Derselbe Baustein ist später Bankkonsortium und Zentralbank — nur größer.

LiquiPool im Detail: Pool steuert alle Bankbeziehungen



Vorher (grau): jede Firma hängt *exklusiv* an ihrer eigenen Bank — drei getrennte Linien, drei Kreditrahmen. **LiquiPool:** alle Firmen melden ihren **Bedarf** an den **LiquiPool-Agenten** (blau). Der zieht **nicht alle Banken gleichzeitig**, sondern jeweils die **günstigste noch freie Fazilität zuerst** (Merit-Order: erst Bank 1, dann 2, dann 3 — nur so viel wie nötig) und verteilt die **Liquidität** (rot), die Ersparnisse werden fair (Shapley) verteilt. **Ergebnis:** weniger gebundenes Kapital, ein gemeinsamer Pool statt drei Inseln — *die Komposition schafft den Wert* ($3 \times 1 \neq 1 \times 3$). **Zudem:** gegenläufige Rechnungen im Konsortium werden *vor* dem Settlement verrechnet — nur der Saldo wird bezahlt. **Und:** überschüssiges Cash wird direkt firmenübergreifend verliehen — *ohne Bank*.

Der nächste Schritt

- **Schnell sichtbarer Nutzen** über LiquiPool, danach Ausbau zu Kostenrechnung, Planung und Restrukturierung.
- **Gemeinsame Co-Entwicklung** an Ihrem konkreten Anwendungsfall — z. B. Ihre Kostenrechnung im OiC.OS abbilden und automatisch als Software ausspielen.
- **Sie behalten einen lernenden, mitlaufenden digitalen Zwilling** Ihrer Organisation — kein Beratungsfoliensatz, sondern ein Asset.

OiC.OS — Organisationen aus Bausteinen. Das System lernt.