

2.– 5. September 2013
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Identitätskrise

Zeit, Zustand und das Reinheitsgebot

Andreas Flierl

imbus AG

Identitätskrise

Zeit, Zustand und das Reinheitsgebot

Der Scherzkeks da vorne...



noch **keine** 10 Jahre im Geschäft

kein Experte für verteilte Systeme, Cloud, NoSQL

kein akademischer Abschluß

noch **nie** Fortran programmiert

noch **kein** vielgenutztes Open-Source-Projekt geschaffen

mit der Gesamtsituation **unzufrieden**

“Conventional programming languages are growing ever more enormous, but not stronger. Inherent defects at the most basic level cause them to be both fat and weak: their primitive word-at-a-time style of programming inherited from their common ancestor--the von Neumann computer, their close coupling of semantics to state transitions, their division of programming into a world of expressions and a world of statements, their inability to effectively use powerful combining forms for building new programs from existing ones, and their lack of useful mathematical properties for reasoning about programs.”

John Backus, 1978

“Conventional programming languages are growing ever more enormous, but not stronger. Inherent defects at the most basic level cause them to be both fat and weak: their primitive word-at-a-time style of programming inherited from their common ancestor--the von Neumann computer, their **close coupling of semantics to state transitions**, their **division of programming into a world of expressions and a world of statements**, their inability to effectively use powerful combining forms for building new programs from existing ones, and their lack of useful mathematical properties for reasoning about programs.”

John Backus, 1978

Sind wir am Ziel?

Sind wir *näher* am Ziel?

State of the art

Position Aug 2013	Position Aug 2012	Delta in Position	Programming Language	Ratings Aug 2013	Delta Aug 2012	Status
1	2	↑	Java	15.978%	-0.37%	A
2	1	↓	C	15.974%	-2.96%	A
3	4	↑	C++	9.371%	+0.04%	A
4	3	↓	Objective-C	8.082%	-1.46%	A
5	6	↑	PHP	6.694%	+1.17%	A
6	5	↓	C#	6.117%	-0.47%	A
7	7	=	(Visual) Basic	3.873%	-1.46%	A
8	8	=	Python	3.603%	-0.27%	A
9	11	↑↑	JavaScript	2.093%	+0.73%	A
10	10	=	Ruby	2.067%	+0.38%	A

State of the art

- Gemeinsamkeiten wichtiger als Unterschiede

State of the art

- Gut genug?
- Konsens?
- Faktische Hinweise?
- OOP “gut”?
- Wohin?

State of the art

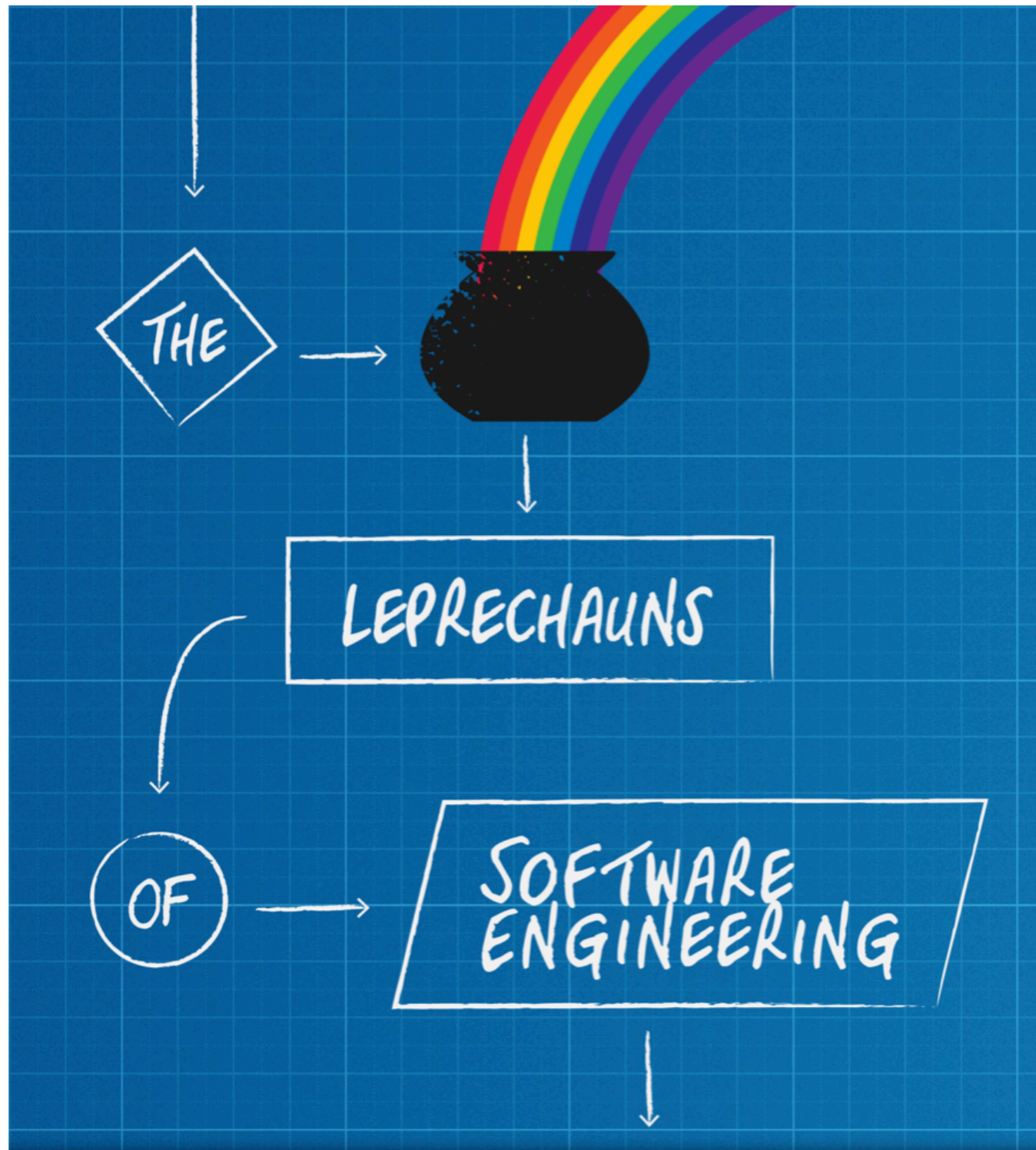
- Nicht am Ziel?
- Falscher Weg?



*“There is nothing more deceptive
than an obvious fact.”*

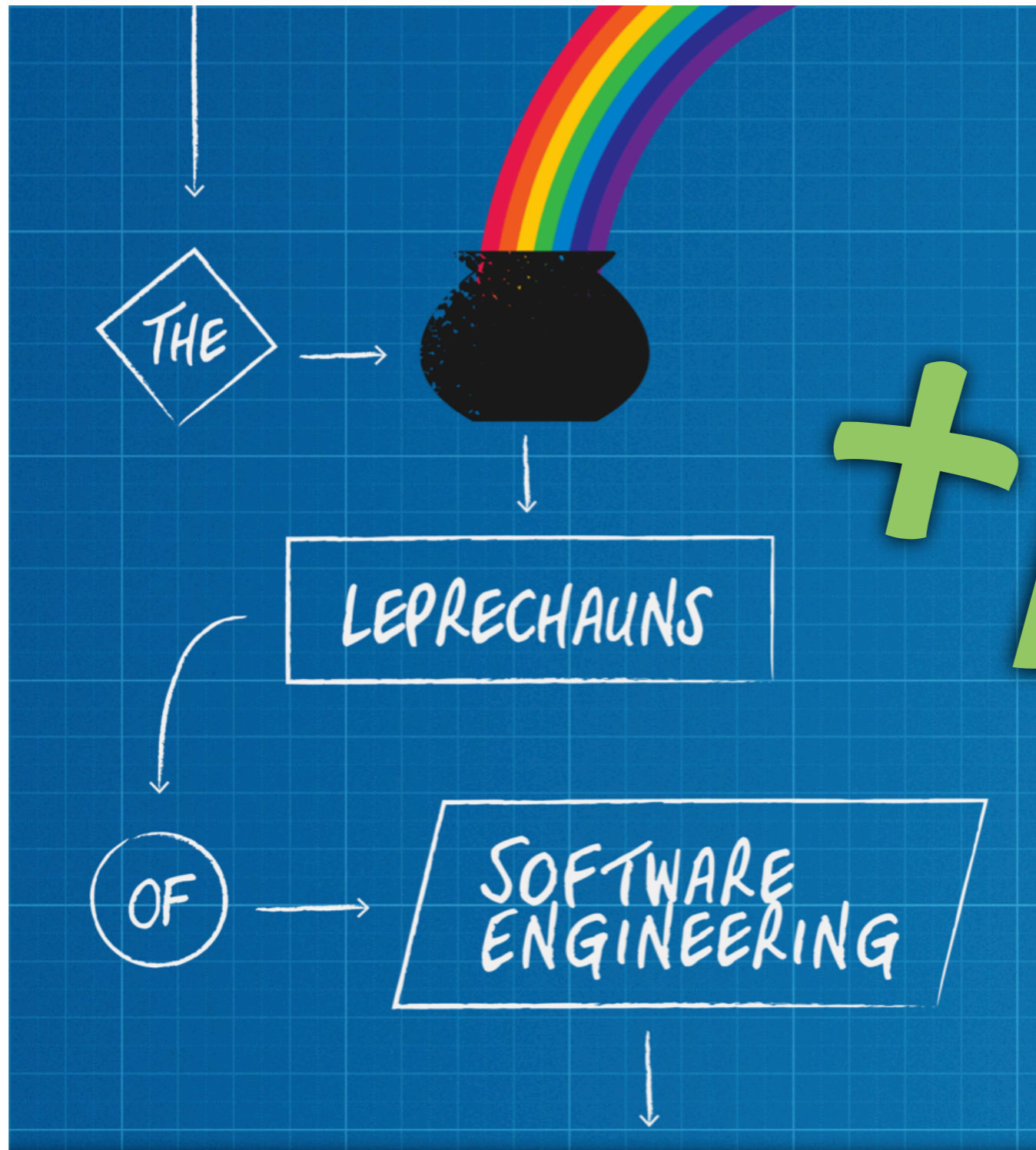
- Sir Arthur Conan Doyle





HOW FOLKLORE TURNS INTO FACT AND WHAT TO DO ABOUT IT

LAURENT BOSSAVIT



HOW FOLKLORE TURNS INTO FACT AND WHAT TO DO ABOUT IT

LAURENT BOSSAVIT

Wortklauberei

Wortklauberei

klau·ben:

[1] etwas mühevoll heraussuchen

[2] pflücken, sammeln

einfach \neq *einfach*

simple \neq *easy*

einfach ≠ leicht

einfach

- simpel: einfältig (von “simplex”)
- eine Rolle, Aufgabe, Konzept, Dimension
- nicht “verwoben”
- nicht komplex

- ist objektiv

einfach

- simpel: einfältig (von “simplex”)
- eine Rolle, Aufgabe, Konzept, Dimension
- nicht “verwoben”
- nicht komplex
- ist objektiv

Vorsicht!

leicht

- naheliegend
 - vertraut
 - verständlich
 - nicht schwer
-
- ist subjektiv, relativ (z.B. Sprachen)

komplex

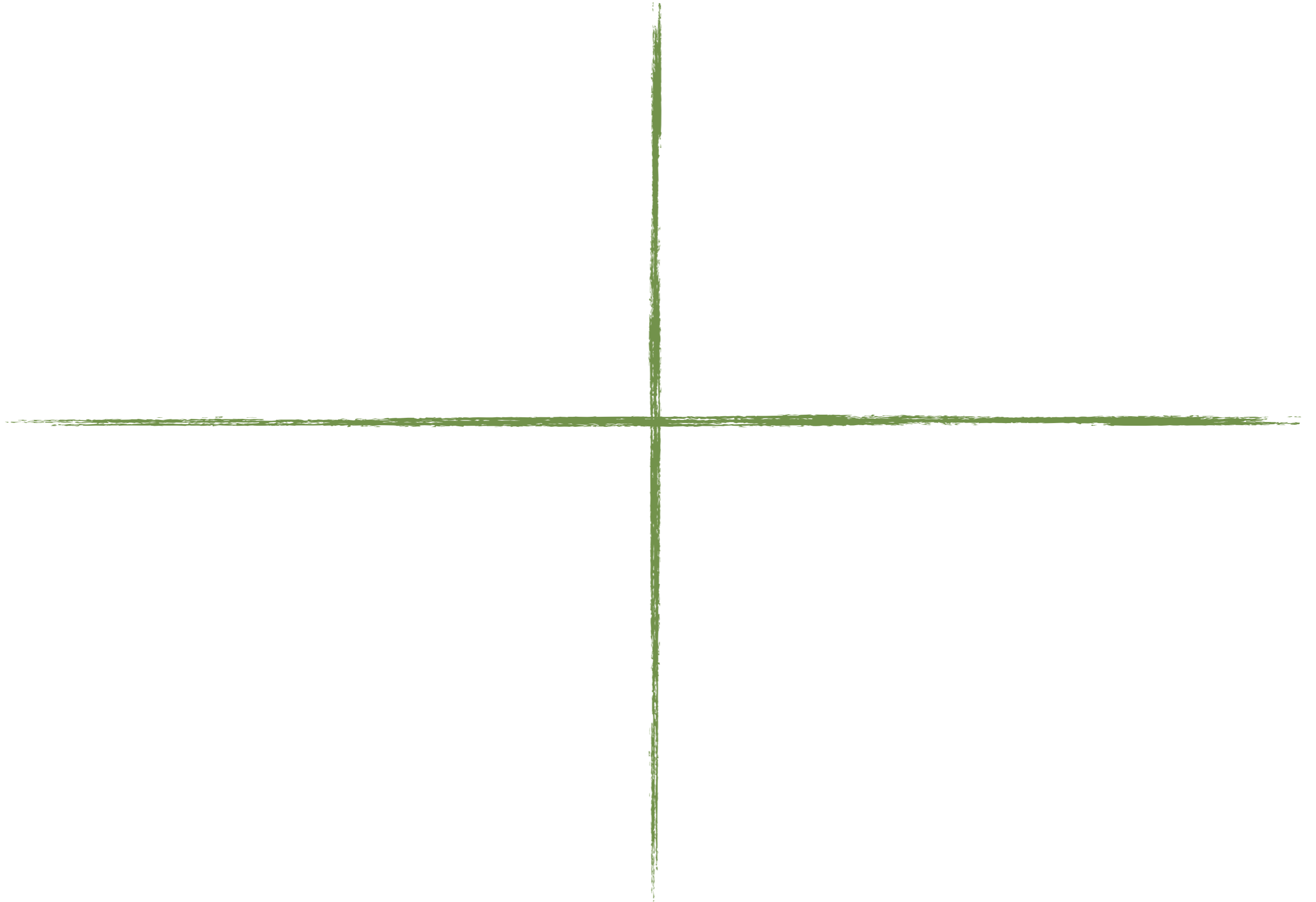
- verflochten, verwoben
- zusammenhängend
- umfassend
- vielschichtig



naheliegend...

- ...zu unseren Fertigkeiten
- unbequemes Thema?
- 7 ± 2
- Verflochtenes muß gemeinsam betrachtet werden
- komplex → meist schwer

noch etwas genauer...



noch etwas genauer...

kompliziert

komplex

einfach

chaotisch

noch etwas genauer...

kompliziert

komplex

einfach

- ein Ding
- eine / wenige Variablen
- lineare Zusammenhänge
- voll vorhersagbar
- planbar

chaotisch

noch etwas genauer...

kompliziert

- viele Dinge
- viele Variablen
- lineare Zusammenhänge
- voll vorhersagbar
- aber aufwendig
- lösbare Probleme

komplex

einfach

- ein Ding
- eine / wenige Variablen
- lineare Zusammenhänge
- voll vorhersagbar
- planbar

chaotisch

noch etwas genauer...

kompliziert

- viele Dinge
- viele Variablen
- lineare Zusammenhänge
- voll vorhersagbar
- aber aufwendig
- lösbare Probleme

komplex

- mehrere, nicht unbedingt viele Dinge
- unbestimmt, evtl. unendl. Variablen
- nicht-lineare Zusammenhänge
- nur in Retrospektive verstehbar
- “Zwickmühlen”
- informiertes Raten, Intuition

einfach

- ein Ding
- eine / wenige Variablen
- lineare Zusammenhänge
- voll vorhersagbar
- planbar

chaotisch

noch etwas genauer...

kompliziert

- viele Dinge
- viele Variablen
- lineare Zusammenhänge
- voll vorhersagbar
- aber aufwendig
- lösbare Probleme

komplex

- mehrere, nicht unbedingt viele Dinge
- unbestimmt, evtl. unendl. Variablen
- nicht-lineare Zusammenhänge
- nur in Retrospektive verstehbar
- “Zwickmühlen”
- informiertes Raten, Intuition

einfach

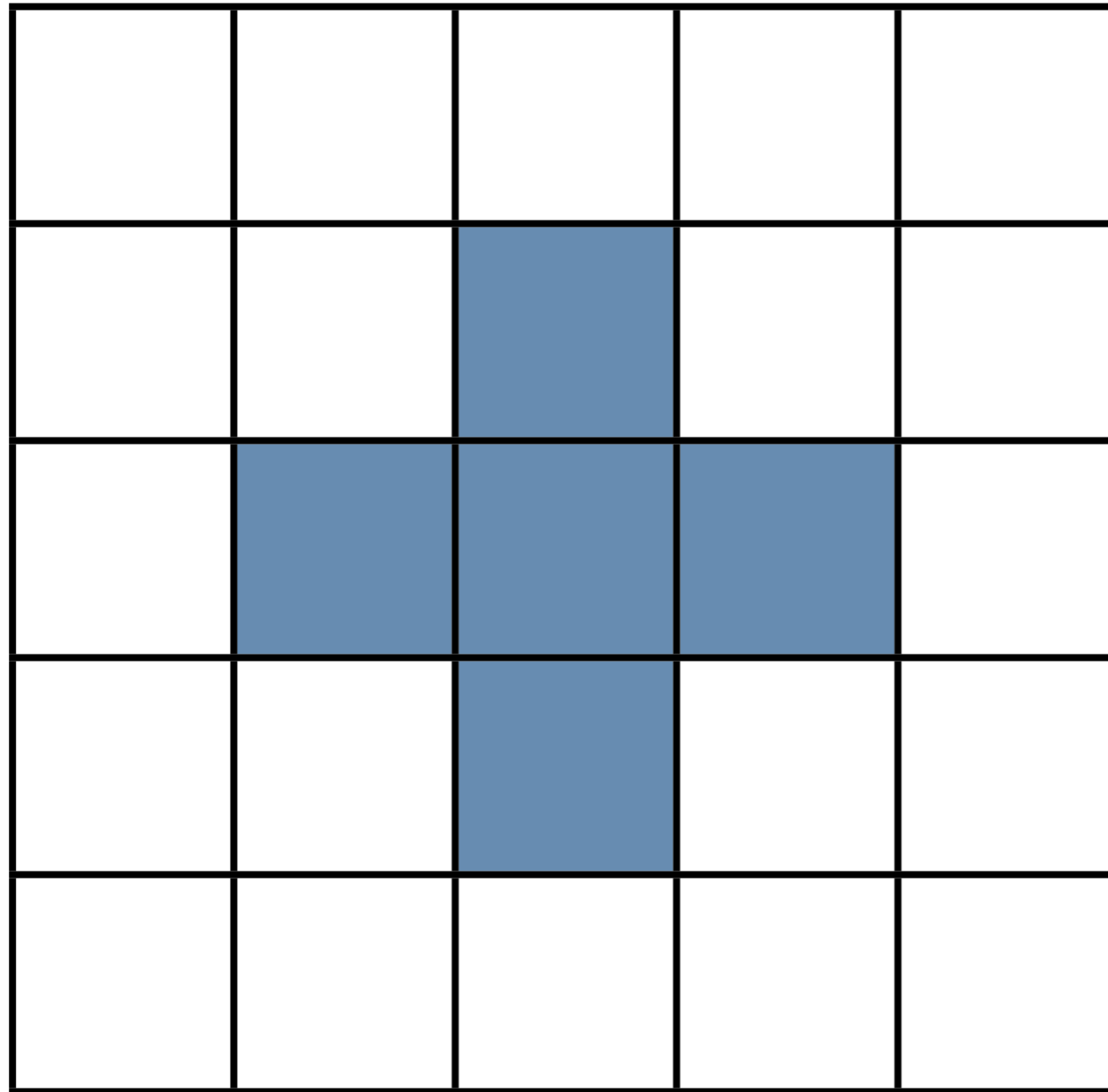
- ein Ding
- eine / wenige Variablen
- lineare Zusammenhänge
- voll vorhersagbar
- planbar

chaotisch

- viele Dinge
- unbestimmt, evtl. unendl. Variablen
- nicht-lineare Zusammenhänge
- auch im Nachgang nicht verstehbar
- keine Vorhersagen oder Schätzungen
- spontane, lokale Reaktion

lights out

lights out



lights out

lights out

■	■			
■				

lights out

		■		
■	■			

lights out

			■	■
	■			
				■
■				

Quellcode kontra Produkt

- Fokus auf leichte Herstellung...
 - leicht für den Programmierer
 - Expressivität
 - Tools
- ...statt auf einfaches Erzeugnis
 - Qualität (Verifikation, Validierung)
 - Wartbarkeit
 - Änderbarkeit

Komplexität

- Notwendig
- Inhärent

- Nebensächlich
- Unbeabsichtigt
- “unfallartig”
- Ballast

Abstraktion

- “aus einem höheren Blickwinkel betrachten”
- Verallgemeinerung
- Vereinfachung
- **nicht** verstecken
- Wer, was, wann, wo, warum, wie?
- “Ich will es nicht wissen.”

Was?

- Tätigkeiten
- Abstraktion durch kleine Mengen verwandter Funktionen
- (Ad-Hoc)-Polymorphie
- Ein-/Ausgabe, Semantik
- benutzt Werte, Abstraktionen

- nicht vermengen mit: “Wie?”

Wer?

- Dinge, die Abstraktionen implementieren
- Zusammensetzen aus Unterkomponenten
- Abhängigkeiten nicht fest kodieren
- nicht mit den Details anderer Dinge vermengen

Wie?

- die eigentliche Implementierung
- Logik
- Inseln
- Polymorphie für offene Systeme
- mit nichts vermengen

Wann? Wo?

- direkte Aufrufe
- stattdessen: Queues
- mit nichts vermengen

Warum?

- Strategien
- Regeln
- meist überall verstreut
- deklarativ, extern

Zwischenstand

- Leichter machen
- Leistungsfähigkeit?
- Vereinfachen
- Bewußte Wahl
 - Vorteile, Nachteile
 - Kompromisse
 - Verantwortung, Professionalität
- Signifikant einfachere
Komponenten

Wert, Identität, Zustand

Wert, Identität, Zustand



Zeit,
Verhalten

Metaphysik und Du

- Was gibt es?
- Wie ist es beschaffen?
- Fundamente, Voraussetzungen, Ursachen
- Sinn und Zweck
- Alfred North Whitehead (1861-1947)
- “Process and Reality”

Metaphysik und Du

- Was gibt es?
- Wie ist es beschaffen?
- Fundamente, Voraussetzungen, Ursachen
- Sinn und Zweck
- Alfred North Whitehead (1861-1947)
- “Process and Reality”

→ Modell

Wert

- unveränderlich
- Zahl
- Größenangabe
- Menge

Wert

- zusammensetzbar
- Menge
- Datum

Wert

- zusammensetzbar
- Menge
- Datum



$\{ n \in \mathbb{N} \mid 1 \leq n \leq 10 \}$

vs.

`myHashSet.add(42)`

Wert

- zusammensetzbar
- Menge
- Datum

5. September 2013

vs.

`myDate.set(Calendar.YEAR, 2001)`

Wert

- zusammensetzbar
- Menge
- Datum



im Gegensatz
zu
Locks

Wert

- Trennung von Veränderung
- vergleichbar
- leicht erzeugbar
- unabhängig von Programmiersprachen

Wert

- Trennung von Veränderung
- vergleichbar
- leicht erzeugbar
- unabhängig von Programmiersprachen

kontra Vererbung
kontra Veränderbarkeit
siehe “Effective Java” Item 8
siehe “Programming in Scala”

Zeit

- wann Dinge passieren
- davor
- danach
- gleichzeitig
- jetzt
- meist relativ
- Ordnung zweier diskreter Dinge

Zeit

- Datum: benannter Zeitpunkt
- Taktgeber
- gemessene Zeitdifferenz
- abgeleitete Größe
- Synchronisation

Identität

- gedankliches Hilfsmittel
- kausal verwandte Werte
- logischer Zusammenhang
- Kontinuität “übergestülpt”
- kein Ort (inkl. Zeit)

Identität

- gedankliches Hilfsmittel
- kausal verwandte Werte
- logischer Zusammenhang
- Kontinuität “übergestülpt”
- kein Ort (inkl. Zeit)



Identität

- gedankliches Hilfsmittel
- kausal verwandte Werte
- logischer Zusammenhang
- Kontinuität “übergestülpt”
- kein Ort (inkl. Zeit)



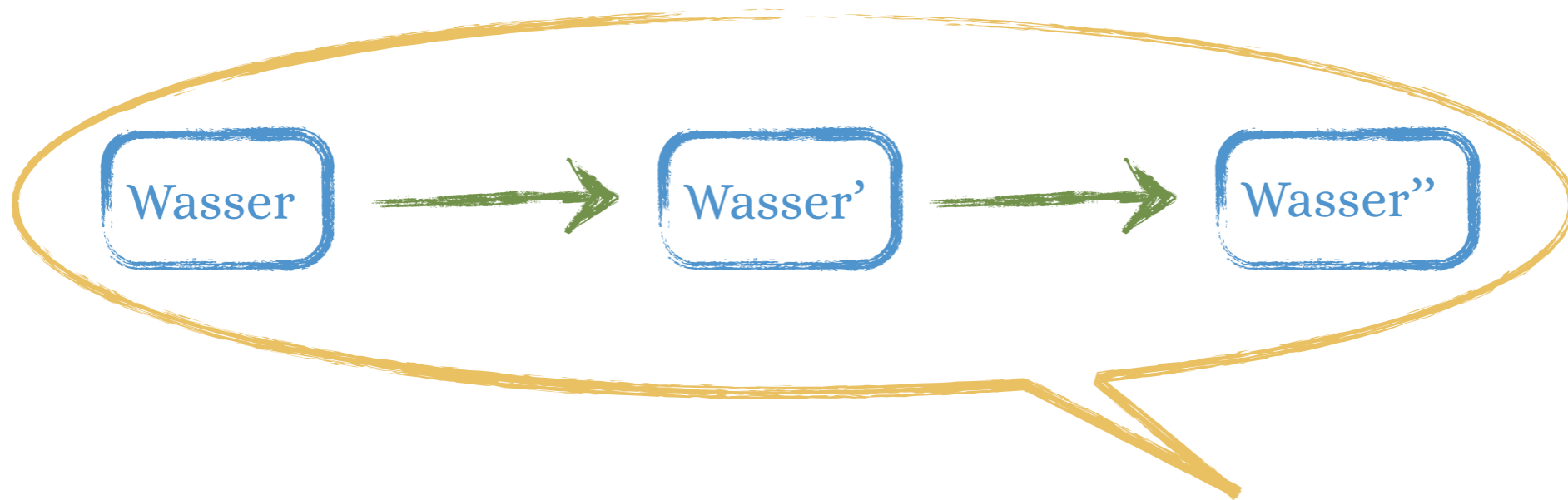
Identität

- gedankliches Hilfsmittel
- kausal verwandte Werte
- logischer Zusammenhang
- Kontinuität “übergestülpt”
- kein Ort (inkl. Zeit)



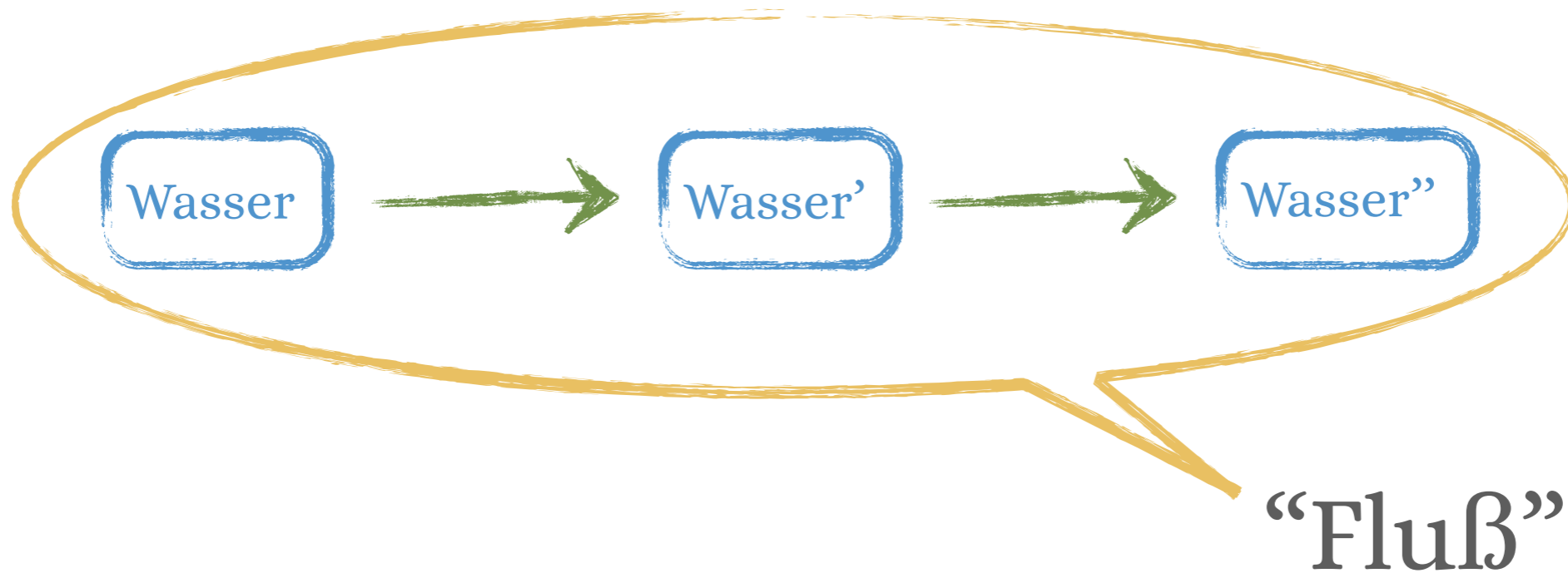
Identität

- gedankliches Hilfsmittel
- kausal verwandte Werte
- logischer Zusammenhang
- Kontinuität “übergestülpt”
- kein Ort (inkl. Zeit)



Identität

- gedankliches Hilfsmittel
- kausal verwandte Werte
- logischer Zusammenhang
- Kontinuität “übergestülpt”
- kein Ort (inkl. Zeit)



Wolke



Wolke



Wolke



Wolke



Wolke(n)?



“The biggest problem we have is that we've conflated 2 things. We've said the idea that I attached to this thing that lasts over time IS the thing that lasts over time. And that's not actually true.”

Rich Hickey

Identität

- kann benannt werden
- nicht das gleiche wie ein Name
- auch Verbund

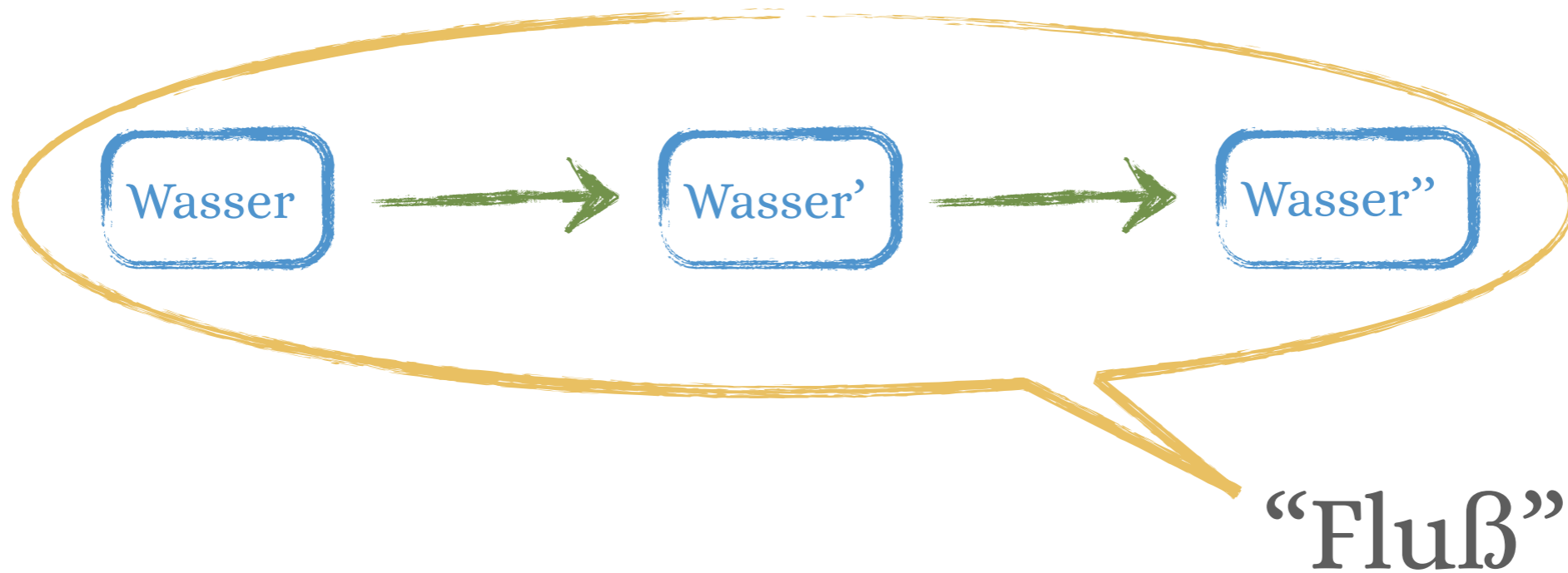
- notwendig für Prozesse
- nicht für Funktionen

Zustand

- Wert einer Identität zu einem Zeitpunkt

Zustand

- Wert einer Identität zu einem Zeitpunkt

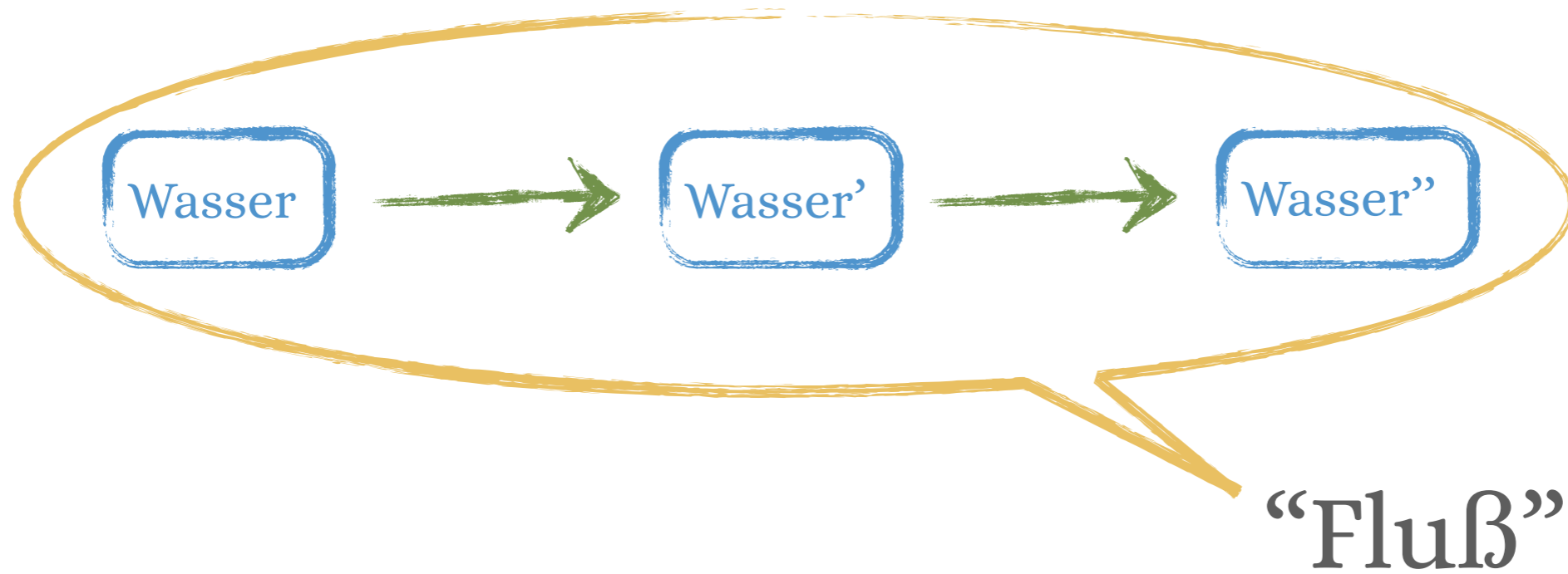


Verhalten

- Wer verhält sich?

Verhalten

- Wer verhält sich?



Warum das alles?

- Entscheidungsfindung
- Stabile Werte nötig
 - wahrnehmbar
 - speicherbar
- Identität nötig
- Richtiger Umgang mit Zustand und Zeit nötig

Wahrnehmung

- Eigenständig
- Losgelöst von Tätigkeit
- Nicht “live”



Wahrnehmung

- Zeit läuft “unerbittlich” weiter
- Debugger
- Locks



Wahrnehmung

- massiv parallel
- unkoordiniert
- kein Austausch von Nachrichten nötig



Umgebung

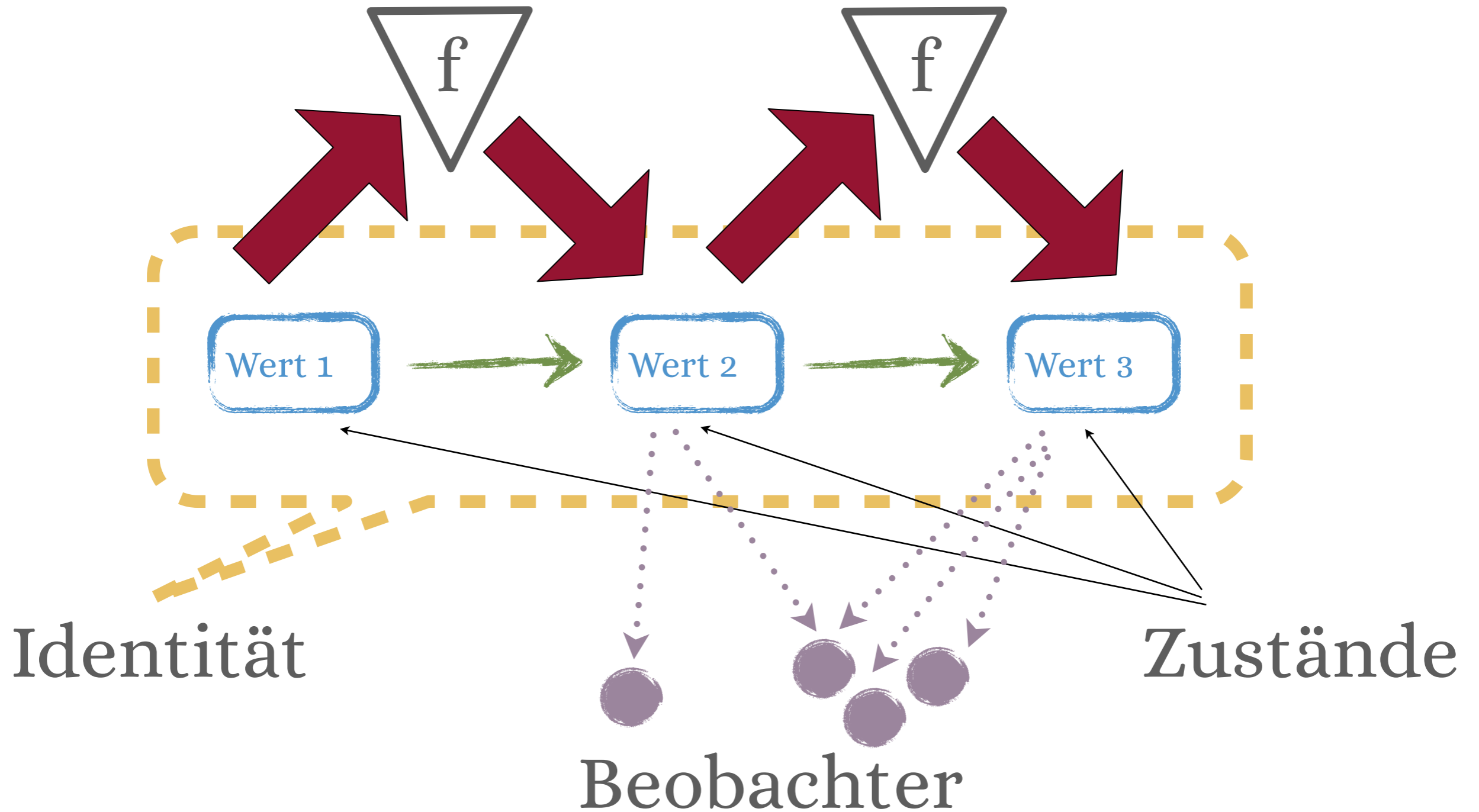
- lokal
- im selben Prozeß
- keine Nachrichten
- bereit für die Frage?
- willens zur Antwort?

- kontra “verteilt”
- “im anderen Gebäude”

Wahrnehmung

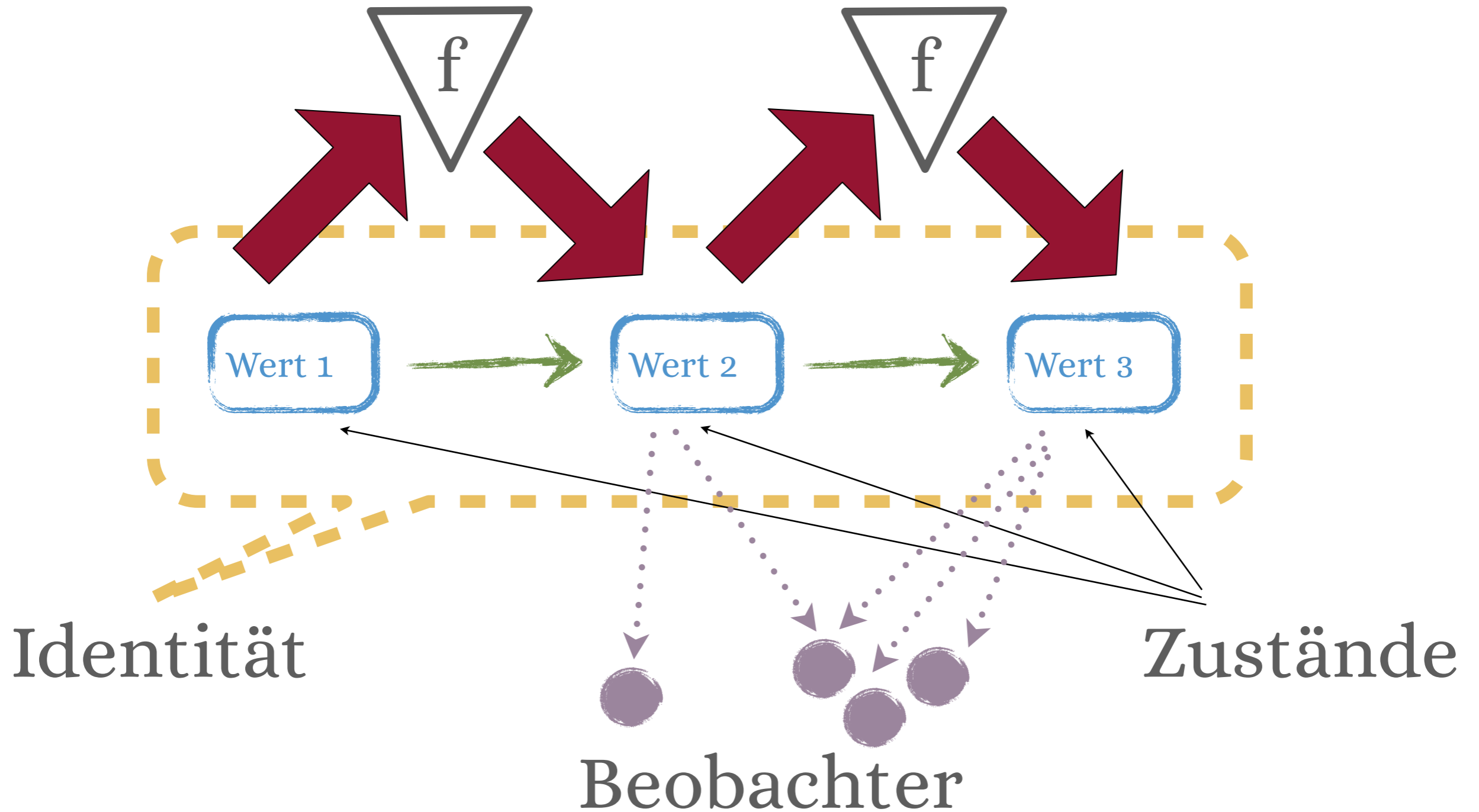
- immer die (unveränderliche) Vergangenheit
- Diskretisierung
- Erkennen von Gleichzeitigkeit
- Schnappschüsse

Auf einen Blick:



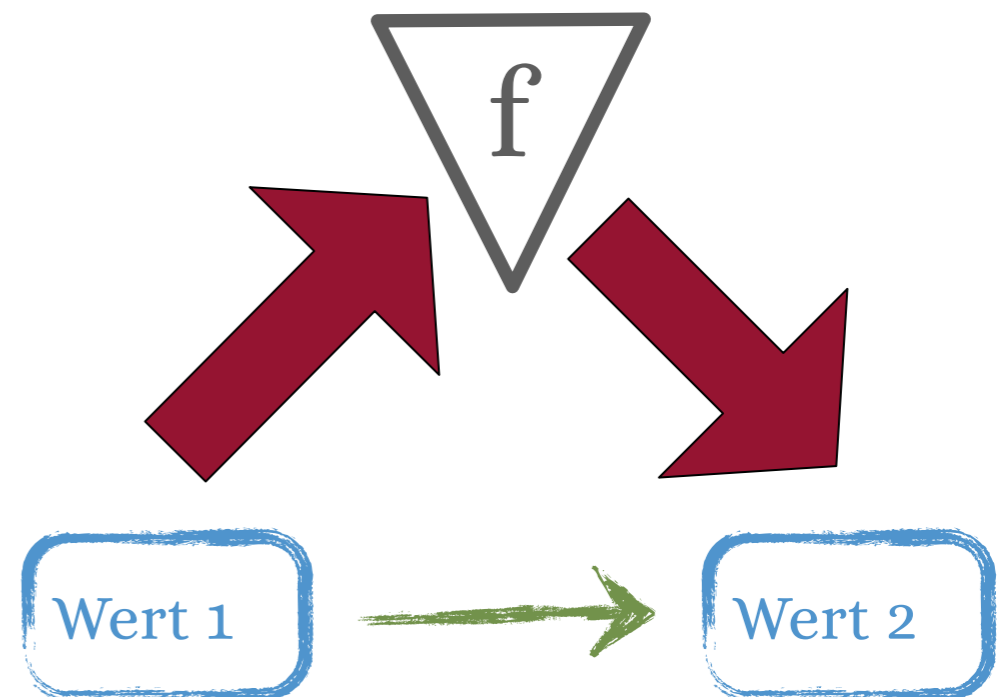
Auf einen Blick:

Funktionen



Implementierung

- Funktion
 - pur, rein
 - keine Nebeneffekte
 - hängt nur von den Argumenten ab
 - lokal
 - kennt keine Zeit (außer Zeit ist Teil der Fachlichkeit)
 - einfach
 - leicht
 - zu verstehen
 - zu ändern
 - zu testen
 - zusammensetzen



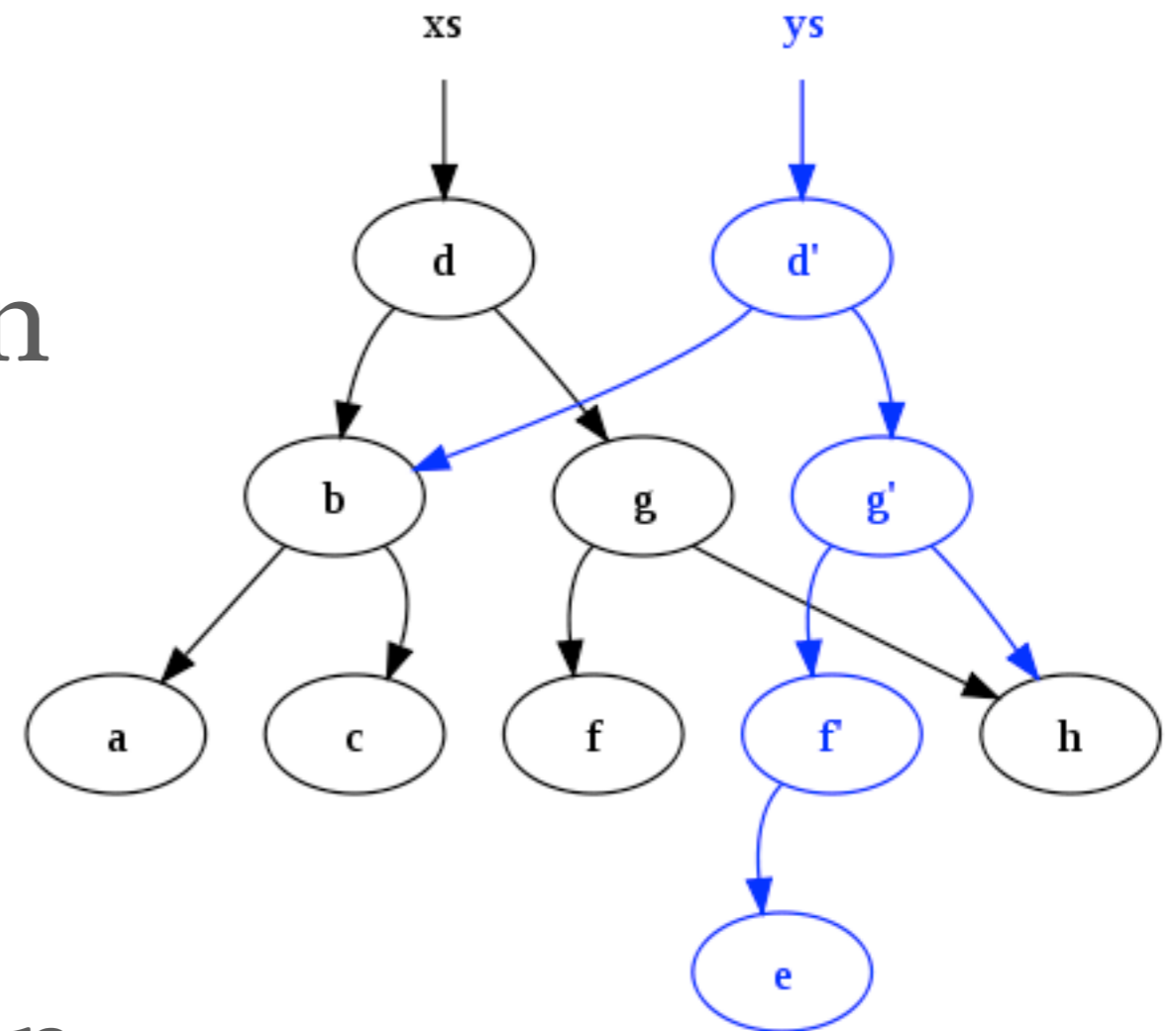
Implementierung

- Werte
 - leicht erzeugbar
 - Erinnerungen
 - teilen / weitergeben
- für Atome:
 - Primitives (int, float etc.)
 - unveränderliche Objekte (value objects)
- für zusammengesetzte Werte:
“persistent data structures”

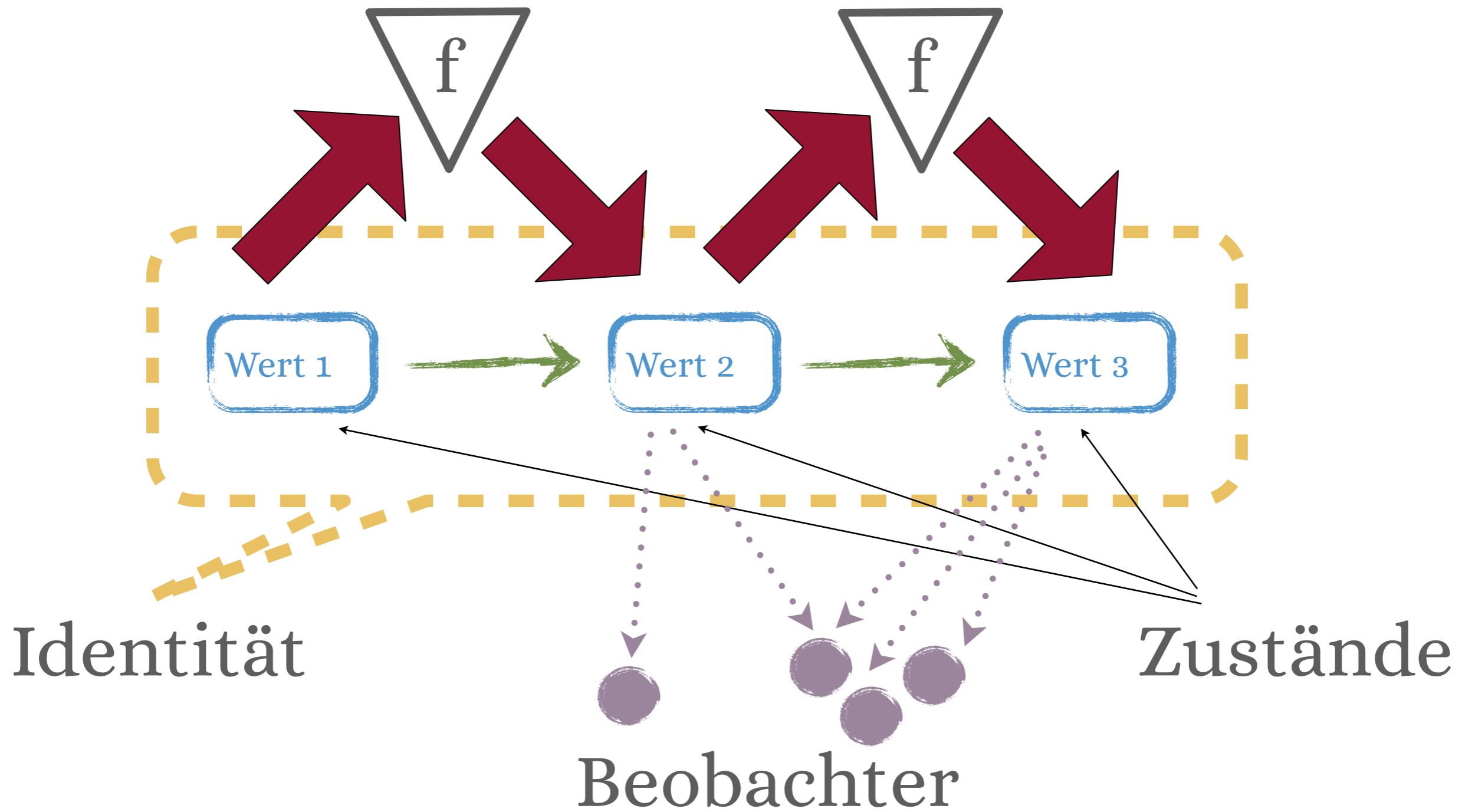
Persistent Data Structures

- unveränderlich
- baumartig
- strukturelles Teilen

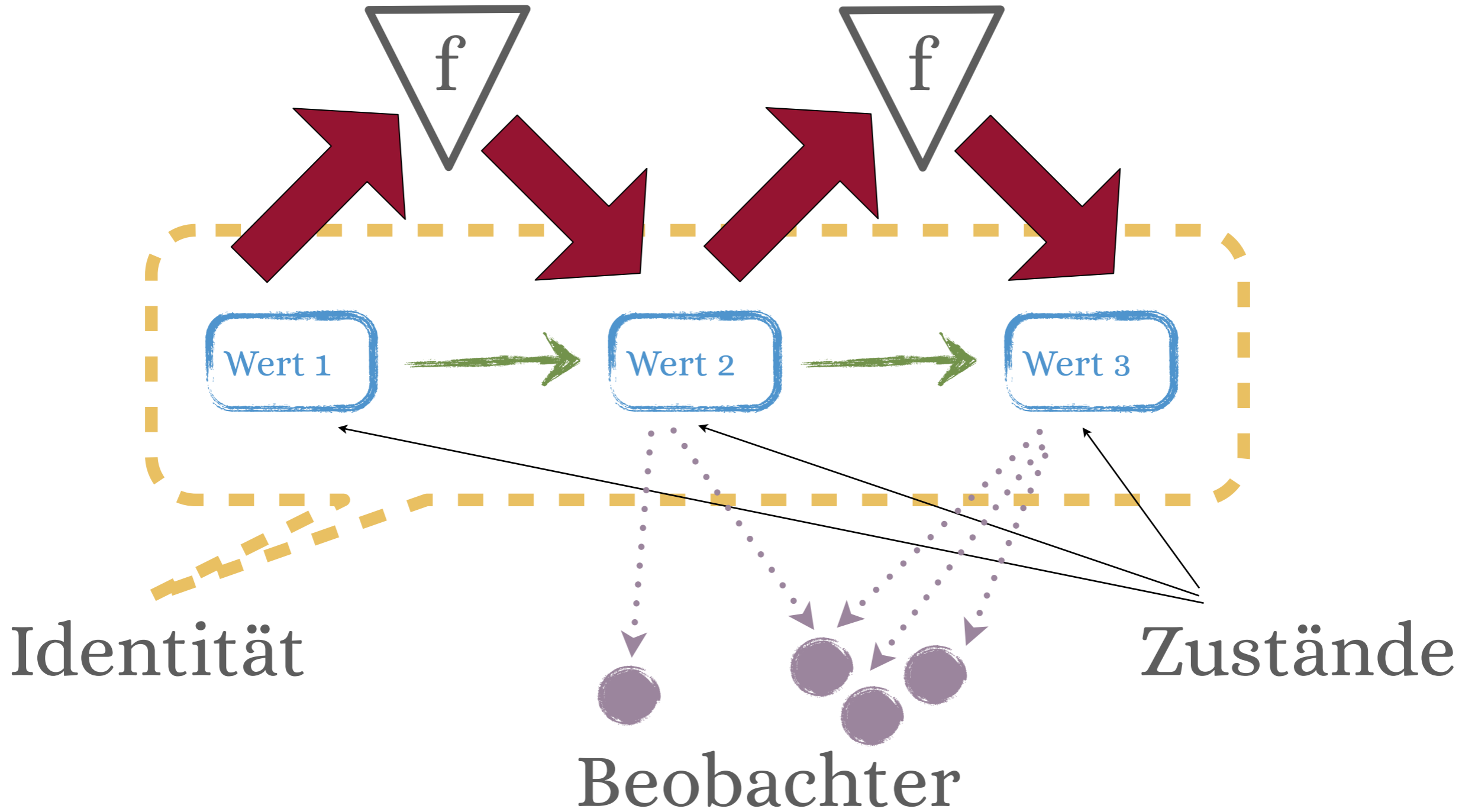
- ziemlich gute Performance



- Implementierungen verfügbar
- Substanzielle Vereinfachung



Funktionen



Implementierung

- Zeit-Koordinierung
 - atomares Aufeinanderfolgen von Werten
 - beobachtbar
 - mehrere Zeitstrahlen
- Compare And Swap
- Agents
- Software Transactional Memory (MVCC)
- Monaden
- Functional Reactive Programming
- ...

Zeit-Koordinierung

- Wesentlich:
 - darum kümmern
 - kann (und muß) es trennen
 - einfachstes ausreichendes Konstrukt wählen



phew

/fyō/

Exclamation

Expressing a strong reaction of relief.

Zusammenfassung

- Namen und Begriffe sind wichtig
- Vertrautheit versteckt Komplexität
- Eine Quelle: unkontrollierte Zustandsübergänge
- Programmieren mit Funktionen und unveränderlichen Werten
- Modell allgemeingültige Lösung für lokale Prozesse
- “there is no silver bullet”

2.– 5. September 2013
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vielen Dank

Andreas Flierl

imbus AG

imbus AG

Spezialisierte Lösungsanbieter für
Software-Qualitätssicherung und Software-Test

Innovativ seit 1992

Erfahrung und Know-how aus mehr als 4.000
erfolgreichen Projekten

Über 200 Mitarbeiter an fünf Standorten in
Deutschland

Beratung, Test-Services, Training, Tools,
Datenqualität

Für den gesamten Software-Lebenszyklus

www.imbus.de

