

# Selmo



## Intelligentes Maschinenverhalten ist eine Entscheidung

---

### Selmo Method™

Maschinenverhalten formal definieren, eindeutig entscheidbar und nachvollziehbar machen.  
Von BlackBox-Denken zu WhiteBox-Verantwortung.

# WhiteBox Engineering: Verhalten wird definiert. Verantwortung wird klar.



BlackBox-Systeme verbergen Verhalten im Code – Risiken werden erst im Betrieb sichtbar. WhiteBox macht Maschinenverhalten vor der Programmierung explizit, prüfbar und entscheidbar. Die Selmo Method verschiebt den Fokus von Hoffnung auf Evidenz – und macht Maschinenverhalten steuerbar.



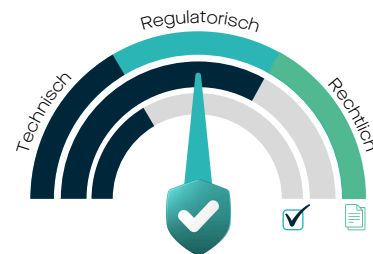
## RoadMap zur WhiteBox

- **01 – Maschinenverhalten entscheiden**  
Gemeinsames Maschinenverhalten definieren und entscheiden, bevor Technik und Code entstehen.
- **02 – WhiteBox entsteht**  
Verhalten wird formal, eindeutig und vollständig modelliert und spezifiziert.
- **03 – Sicherheit vor Implementierung**  
Simulation prüft Verhalten vor IBN – Risiken werden sichtbar, bevor es teuer wird.
- **04 – Technik folgt der Entscheidung**  
Code wird aus dem Modell abgeleitet – deterministisch statt interpretiert.
- **05 – WhiteBox als Asset**  
Verhalten bleibt auditierbar und stabil – auch bei Änderungen im Lebenszyklus.

**Maschinenverhalten unter Kontrolle. Ihre Maschine ist nun ein Premium Asset.**

## Risiko wird steuerbar

### Governance vor Inbetriebnahme



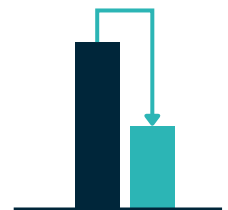
Technische, regulatorische und rechtliche Risiken werden bereits im Modell gemanagt – lange vor dem physischen Betrieb.

### BlackBox vs. WhiteBox Kosten

WhiteBox reduziert Stillstände, IBN-Zeit und Wartung signifikant. BlackBox verlagert Kosten vom Projekt in den Betrieb – und erhöht das Risiko späterer Eskalationen.

## 120 % ROI

Strukturelle Kostenreduktion über den Lebenszyklus.



### WhiteBox Engineering ist Governance

Selmo entscheidet Verhalten vor dem Schreiben von Code – und macht Maschinen zu transparenten, auditierbaren Assets.

WhiteBox ist keine Option. Sie ist die Voraussetzung für kontrollierbaren Betrieb.



# Undefiniertes Maschinenverhalten ist kein Engineering-Problem. Es ist ein Problem der Entscheidungsfindung.

Seit Jahrzehnten werden industrielle Maschinen gebaut, um zu funktionieren – nicht so, dass man ihr Verhalten versteht.

Verhalten entsteht aus Code, Entscheidungen während der Inbetriebnahme, der subjektiven Interpretation von Lieferanten und nachträglichen Korrekturen. Das läuft eine Zeit lang gut, bis es plötzlich Probleme macht.

Was fehlt, ist formale Entscheidungsfindung über das Verhalten.

Das passiert, wenn Maschinenverhalten nicht explizit definiert ist:

- Zustände existieren, die niemand vollständig beschreiben kann
- Übergänge treten auf, die nie formal geprüft werden
- Änderungen häufen sich ohne Validierung durch die Stakeholder

Das Ergebnis ist nicht nur Ineffizienz. Das Ergebnis ist strukturelle Unsicherheit. Entscheidungen zu Sicherheit, Performance, Compliance und zukünftiger Skalierbarkeit werden getroffen, ohne dass klar ist, wie sich die Maschine tatsächlich verhalten soll.

Wenn Verhalten nicht formal entschieden werden kann, kann die Maschine nicht verantwortungsvoll betrieben werden.





# Wenn niemand erklären kann, warum sich die Maschine so verhält, hat auch niemand sie wirklich unter Kontrolle.

Viele Organisationen gehen davon aus, dass sie das Maschinenverhalten kontrollieren. In der Realität ist Kontrolle oft verteilt.

Verhalten wird durch Teams, Lieferanten und kontinuierliche Änderungen geprägt.  
Verantwortung existiert, jedoch ohne einen eindeutigen Eigentümer.



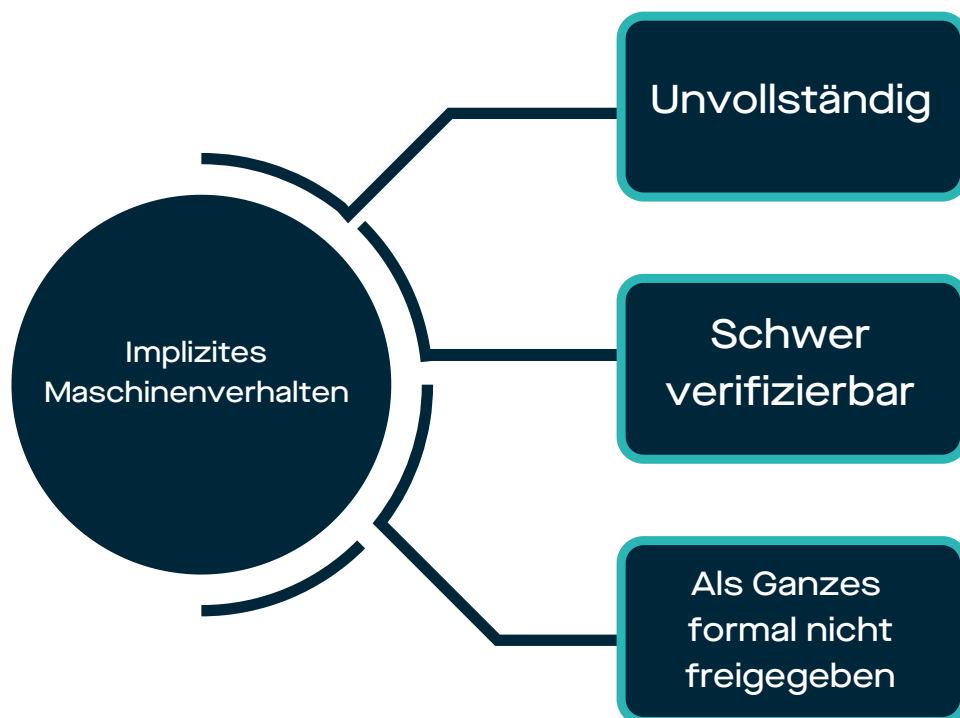
# Über Maschinenverhalten muss vor der Implementierung entschieden werden

Im klassischen Engineering wird Maschinenverhalten zu spät festgelegt.

Wie sich eine Maschine verhält, entsteht erst während Programmierung, Inbetriebnahme und iterativen Fehlersuche, also nachdem mechanische, elektrische und vertragliche Entscheidungen längst getroffen sind.

Zu diesem Zeitpunkt können keine Entscheidungen mehr zum Verhalten getroffen werden, es kann nur noch angepasst werden.

Selbst die erfahrensten Ingenieure, die besten Werkzeuge und die strengsten Reviews können diese strukturelle Tatsache nicht ändern: Code beschreibt, wie ein System ausgeführt wird, nicht warum es sich so verhält. Solange Verhalten implizit über die Implementierung definiert wird, bleibt es:



Wenn erst im Code entschieden wird, wie sich eine Maschine verhält, kann man dieses Verhalten nicht mehr vorher festlegen, man entdeckt es erst im Betrieb.

The background of the slide is a faded, light blue image of an industrial robotic arm. The arm is composed of various metal components, including a gripper at the end, and is surrounded by cables and structural frames. The overall scene is a typical industrial manufacturing environment.

# Die Selmo Method™ definiert genau diese Entscheidungsebene.

Sie etabliert ein formales, deterministisches Modell, in dem Maschinenverhalten vollständig beschrieben, validiert und freigegeben wird, bevor Ausführungslogik erzeugt wird.

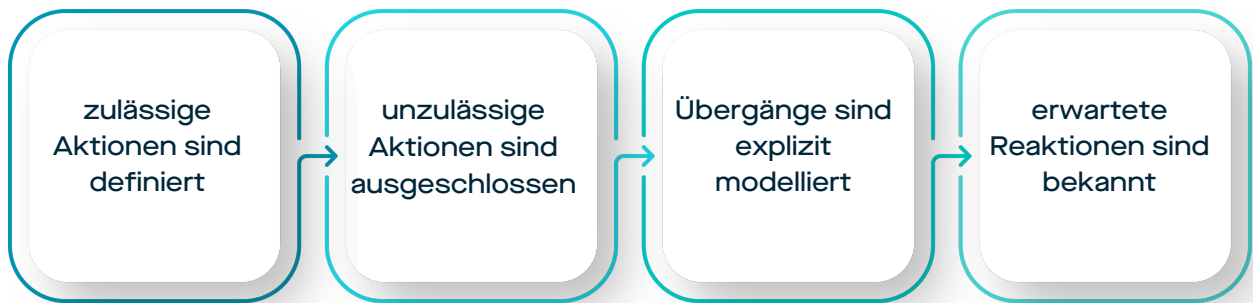
Selmo verbessert nicht das Programmieren. Es entkoppelt Verhalten vom Code.



# Formales Verhalten bedeutet, dass nichts undefiniert bleibt.

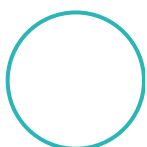
Ein formal festgelegtes Maschinenverhalten bedeutet, dass die Maschine nur explizit definierte Zustände annehmen kann.

Für jeden Zustand gilt:



Es gibt keine versteckten Pfade. Keine impliziten Annahmen.  
Kein Verhalten, das „meistens“ funktioniert.

**Jeder Übergang ist beabsichtigt.**  
**Jeder Zustand ist überprüfbar.**  
**Jedes Verhalten ist erklärbar.**



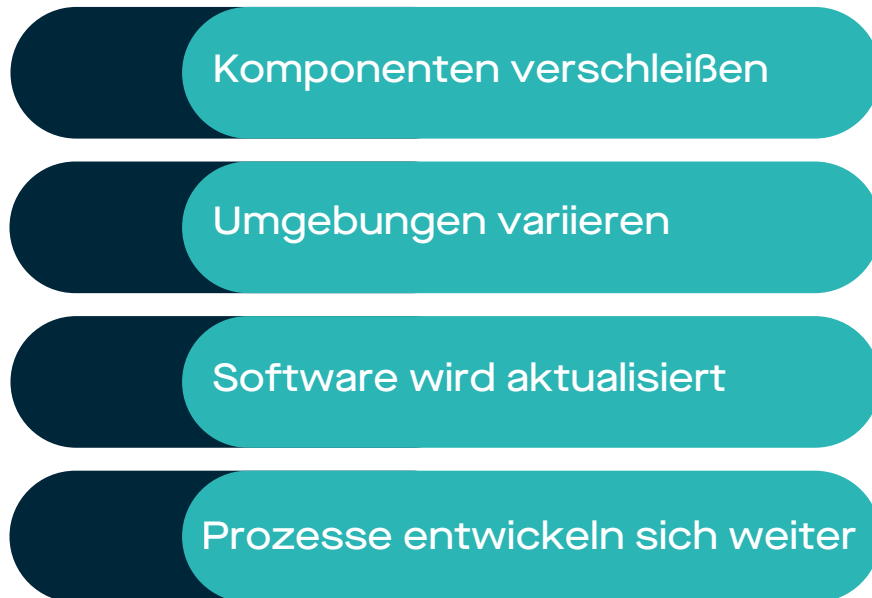
Das ist es, was Determinismus bedeutet, nicht Vorhersagbarkeit durch Erfahrung, sondern Vorhersagbarkeit durch Definition.

Wenn ein Zustand nicht formal beschrieben werden kann,  
darf er nicht existieren.

# Entschiedenes Verhalten muss kontinuierlich verifiziert werden.

Ein formal definiertes Verhaltensmodell ist nur dann sinnvoll, wenn es im Betrieb gültig bleibt.

Maschinen verändern sich:



Ohne regelmäßige Überprüfung entfernt sich selbst klar definiertes Verhalten mit der Zeit von dem, was ursprünglich vorgesehen war.

Damit Maschinen sich wirklich deterministisch verhalten, muss ständig abgeglichen werden,

ob der aktuelle Zustand noch dem entspricht, was vorher festgelegt wurde.

Abweichungen werden dann sofort sichtbar, nicht erst Wochen später bei der Fehlersuche.

So wird der Betrieb transparent:

Verhalten wird nicht angenommen oder vermutet, sondern laufend überprüft und bestätigt.

Verhalten ist nur dann deterministisch,  
wenn Abweichungen sofort erkannt werden.



# Die meisten Probleme im Betrieb sind Symptome, keine Ursachen.

Über industrielle Unternehmen hinweg treten immer wieder die gleichen Frustrationen auf:

“

„Wir wissen nicht wirklich, was die Maschine tut.“

„Das Verhalten ändert sich je nachdem, wer den Code angefasst hat.“

”

„Kleine Änderungen machen an anderer Stelle etwas kaputt.“

”

“

„Diagnosen sind unklar und inkonsistent.“

**Das sind keine isolierten Probleme.** Es sind wiederkehrende Symptome. Was Sie gemeinsam haben, ist nicht eine mangelnde Ausführung, sondern eine fehlende Referenz, wie sich die Maschine verhalten soll.

Ohne ein formal entschiedenes Modell:

- Abweichungen bleiben unentdeckt
- Verhalten wird interpretiert, nicht verifiziert
- Fehlersuche ersetzt Verständnis

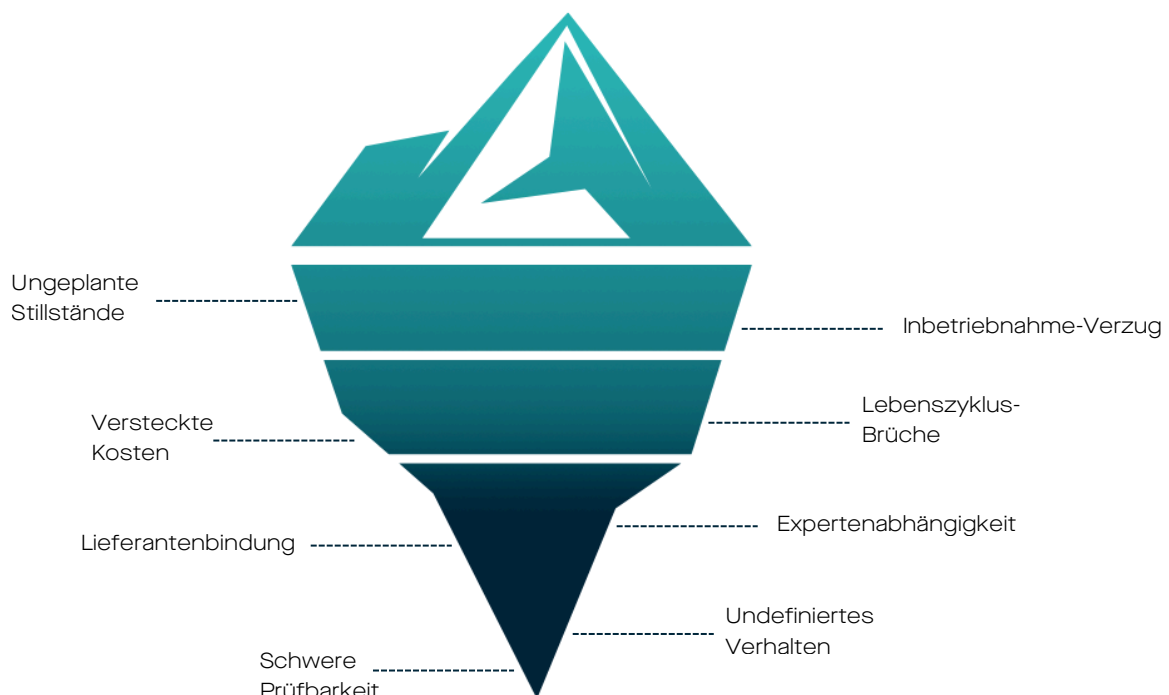
Das System versagt nicht zufällig. Es verhält sich so, weil über das Verhalten nie explizit entschieden wurde.

Wenn Abweichungen strukturell nicht sichtbar sind, werden Probleme immer als Überraschungen auftreten.

# Unentdeckte Abweichungen erzeugen versteckte Risiken

Über industrielle Organisationen hinweg treten immer wieder die gleichen Frustrationen auf. Wenn Abweichungen vom vorgesehenen Verhalten unbemerkt bleiben, summieren sich ihre Auswirkungen. Im Betrieb zeigt sich das als:

- ungeplante Stillstände
- fragile Prozesse
- inkonsistente Performance
- wachsende Abhängigkeit von einzelnen Experten



**Strukturell reichen die Konsequenzen tiefer.**

Entscheidungen zu Sicherheit, Compliance, Durchsatz und zukünftigen Änderungen werden ohne eine verlässliche Grundlage für das Maschinenverhalten getroffen.

Risiken werden nicht bewertet, sie werden absorbiert. Über die Zeit passen sich Organisationen an Unsicherheit an, anstatt sie zu eliminieren. Was sich wie Stabilität anfühlt, ist oft nur das Ausbleiben eines akuten Fehlers.

Risiko entsteht nicht durch Abweichung an sich, sondern durch Abweichung, die unerkant bleibt.



# Premium-Performance erfordert klar definiertes Verhalten.

Industrielle Organisationen investieren stark in Performance, Qualität und Verfügbarkeit.

Sie spezifizieren Toleranzen, Zykluszeiten, Sicherheitsstufen und vertragliche Zusagen, oft bis ins kleinste Detail.

Dennoch bleibt der grundlegendste Aspekt häufig implizit: wie sich die Maschine verhalten darf.

Ohne formale Entscheidung über das Verhalten:



Performance hängt  
von Interpretation ab



Stabilität hängt von  
Einzelpersonen ab



Qualität nimmt über  
die Zeit ab

Premium-Ausführung kann nicht auf undefiniertem Verhalten aufbauen. Sie erfordert Verhalten, das:



explizit definiert und  
entschieden ist



strukturell  
durchgesetzt ist



kontinuierlich  
verifiziert wird

Nur Maschinen mit entschiedenem und definiertem Verhalten können als Premium-Assets geliefert werden.

# Ein vollständiges System für entscheidbares Maschinenverhalten

Die Selmo Method™ integriert acht strukturelle Dimensionen in ein einzelnes deterministisches Framework:

## Deterministische Zustandsarchitektur



Alle Zustände und Übergänge sind konfliktfrei und explizit zugelassen.

## Vorgelagerte Verhaltensverifikation



Verhalten wird validiert, bevor die Ausführung beginnt.

## Automatische Logikableitung

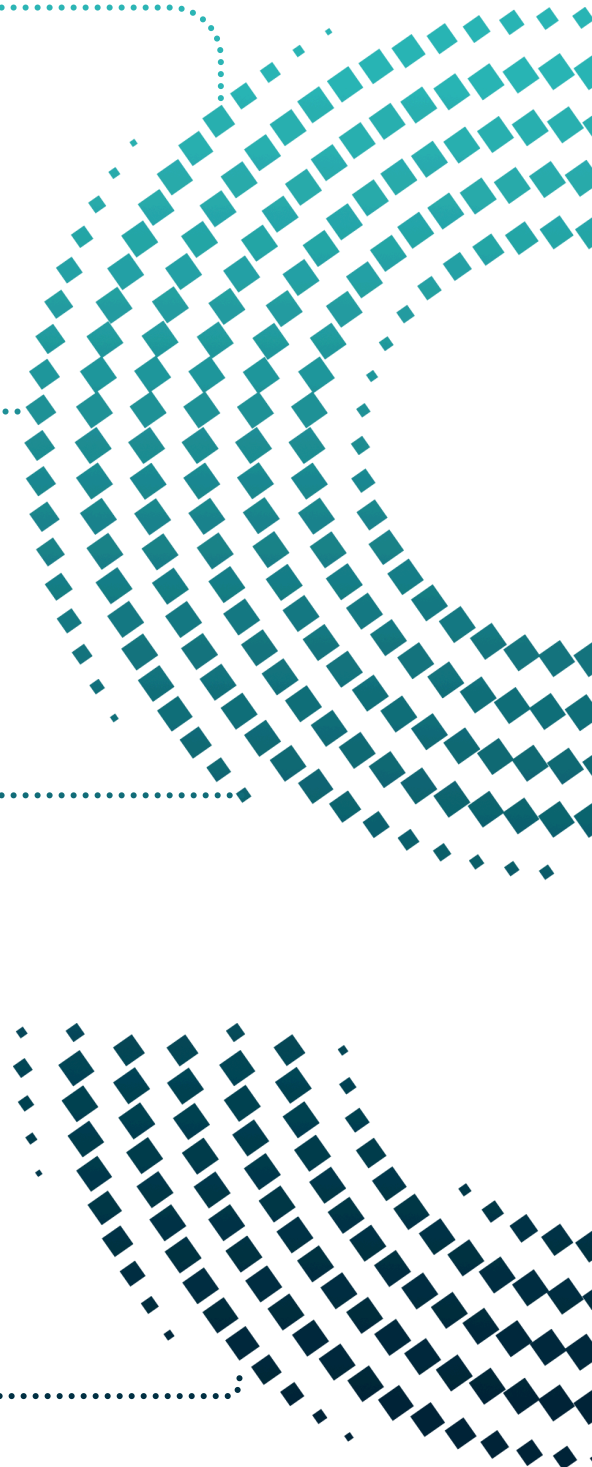


Die Implementierung ist das direkte Ergebnis des zuvor festgelegten Verhaltens.

## Strukturelle Nachverfolgbarkeit



Jede Entscheidung bleibt vom Intent bis zum Betrieb sichtbar.



Entscheidbares Maschinenverhalten lässt sich nicht durch voneinander getrennte Maßnahmen erreichen. Es erfordert ein geschlossenes, kohärentes System, das Verhalten definiert, Korrektheit validiert und Entscheidungen über den gesamten Lebenszyklus hinweg erhält. Genau dafür ist die Selmo Method™ aufgebaut.



### Formale Verhaltensmodellierung

Maschinenverhalten wird mathematisch definiert, nicht informell beschrieben.



### Lebenszyklusstabilität

Verhalten bleibt trotz Änderungen, Updates und Erweiterungen korrekt.



### Deterministische Betriebsdaten

Jedes Signal hat Kontext, Bedeutung und Referenz.



### Organisatorische Standardisierung

Eine gemeinsame Sprache für Maschinenverhalten über Teams und Zulieferer hinweg.

Nur ein durchgängiges System stellt sicher, dass sich Maschinen über Projekte, Zulieferer und Jahre hinweg so verhalten, wie es festgelegt und entschieden wurde.



# Selmo Method™

## Ein formaler Entscheidungsstandard für Maschinenverhalten

Die Selmo Method™ etabliert einen formalen Entscheidungsrahmen, in dem Maschinenverhalten vollständig definiert, geprüft und validiert wird, bevor die Implementierung beginnt.

Sie trennt Verhaltensentscheidungen von der technischen Ausführung.

Dadurch wird Maschinenverhalten zu einem strukturellen, überprüfbaren und auditierbaren Asset.

## Beschreibung unseres Systems

Die Selmo Method™ basiert auf einem deterministischen System. Es zeichnet sich aus durch:

### Formale Verhaltensmodellierung

Jeder Zustand, jede Aktion und jeder Übergang ist explizit definiert.

### Deterministische Zustandsarchitektur

Das Modell ist vollständig konsistent, ohne verborgene Pfade oder undefinierte Zustände.

### Verhaltensverifikation vor der Ausführung

Das vollständige Verhalten wird als korrekt validiert, bevor es in Steuerungslogik implementiert wird.

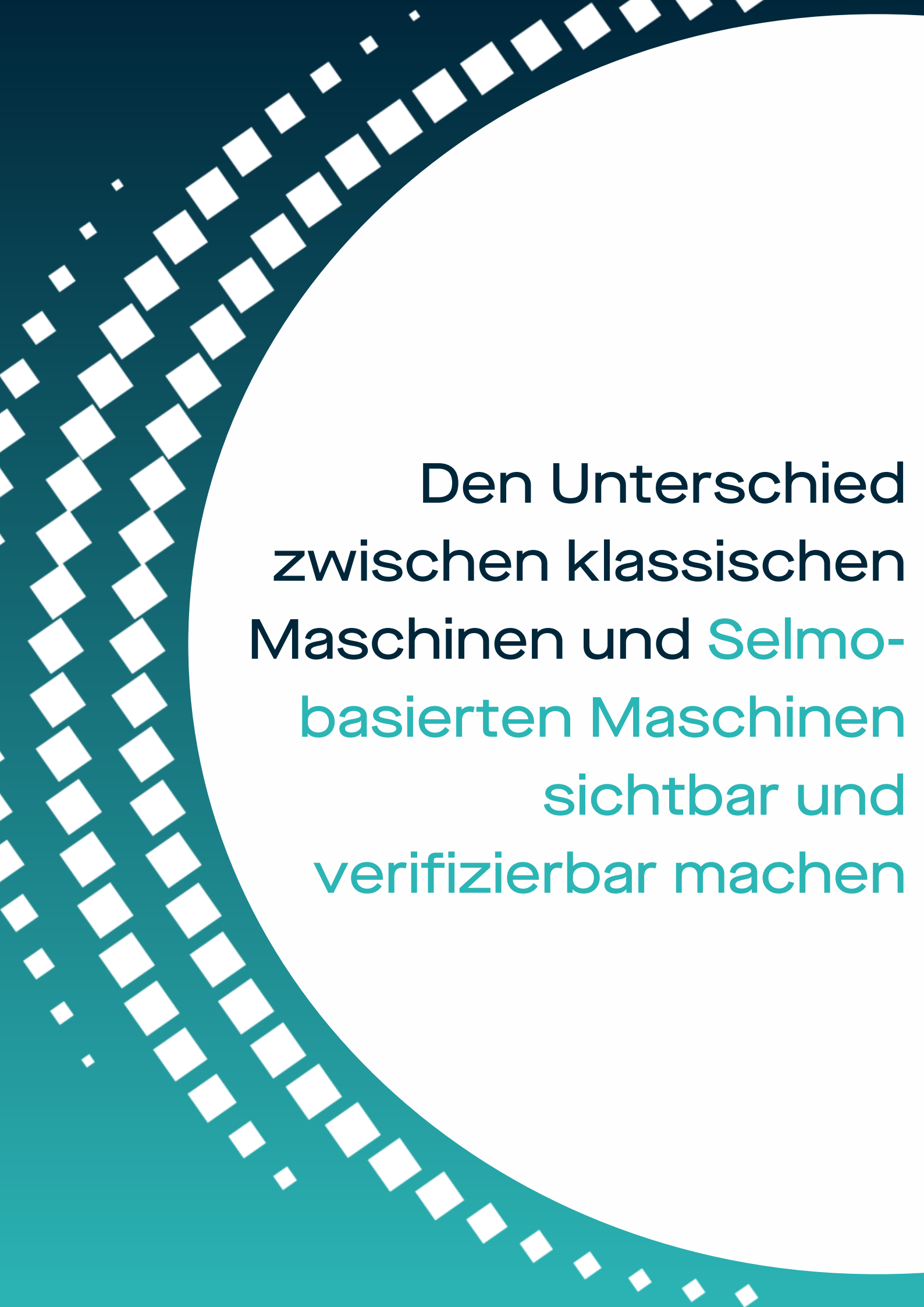
### Strukturelle Nachverfolgbarkeit

Jede Entscheidung ist vom Intent bis zur Ausführung und zum Betrieb nachvollziehbar.

### Standardisiertes Verhalten über Organisationen hinweg

Die gleiche Verhaltenssprache gilt über Teams, Standorte und Lieferanten hinweg.

Selmo optimiert Verhalten nicht.  
Es entscheidet es.

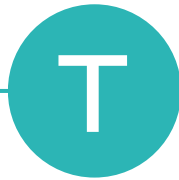


Den Unterschied  
zwischen klassischen  
Maschinen und Selmo-  
basierten Maschinen  
sichtbar und  
verifizierbar machen



### Prozess

Beschreiben Sie, was das System tun soll, als Abfolge von Zuständen, Aktionen und erwarteten Reaktionen.



### Technologie

Festlegen, wie jede erwartete Reaktion physisch mit Sensoren und Aktoren erfasst werden kann.



### Funktion

Die definierte Prozesslogik (P) mit der messbaren Technologie (T) verknüpfen, um die Realisierbarkeit sicherzustellen.



# Von BlackBox-Denken zu WhiteBox-Verantwortung.



Eine riskante Maschine, die als „BlackBox“ arbeitet, in eine transparente „WhiteBox“-Maschine transformieren, in der sämtliches Verhalten explizit und nachvollziehbar ist.

## Automatisierte Codegenerierung

Das Modell wird automatisch in SPS-Code (z. B. Kontaktplan) übersetzt, wodurch manuelle Programmierung entfällt.

## Inbetriebnahme & Validierung

Da die Inbetriebnahme nicht der Problemlösung dient, ist sie ein formaler Abgleich des Modells mit der realen Hardware.

## Formale Modellierung

Die PTF-Ergebnisse werden in ein formales Prozessmodell überführt, das zur einzigen maßgeblichen Referenz wird.

## Live

### „WhiteBox“-Betrieb

Das System betreibt permanent den modellbasiert definierten Zustand mit realen Systemdaten und erkennt jede Abweichung unmittelbar.



# Eine Struktur. Zwei grundlegend unterschiedliche Realitäten.

Maschinen mögen äußerlich identisch erscheinen. Ihre strukturelle Logik ist es nicht.

|              | Konventionelle Maschinen                                                                                                          | Selmo-Maschinen                                                                                                                         |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Verhalten    |  Verhalten entsteht aus der Implementierung      |  Verhalten wird vor Implementierung entschieden       |
| Zustände     |  Zustände sind implizit oder unvollständig       |  alle Zustände und Übergänge sind explizit            |
| Abweichungen |  Abweichungen werden spät oder indirekt erkannt |  Abweichungen werden sofort erkannt                  |
| Wissen       |  Wissen liegt bei Einzelpersonen               |  Wissen ist strukturell verankert                   |
| Änderungen   |  Änderungen führen zu Unsicherheit             |  Änderungen werden gegen das Modell geprüft         |
| Qualität     |  Die Verhaltensqualität nimmt über die Zeit ab |  die Verhaltensqualität bleibt langfristig erhalten |

Der Unterschied liegt nicht in Komponenten oder Code, sondern darin, wie Verhalten definiert, verifiziert und erhalten wird.

Selmo verbessert die Ausführung nicht.  
Es verändert die dahinterliegende Struktur.

# Klarheit vor der Implementierung. Struktur vor der Ausführung.

Entscheidbares Maschinenverhalten entsteht nicht zufällig. Es ist das Ergebnis eines klar definierten Entscheidungsprozesses, der vom Intent bis zur Ausführung führt und dabei Struktur und Verantwortung wahrt.

Die Selmo Method™ etabliert diesen Prozess.

## Der Selmo-Entscheidungsprozess



### PTF – Prozess · Technologie · Funktion

Das gewünschte Maschinenverhalten wird definiert, bevor eine Lösung umgesetzt wird. Prozesse werden geklärt, Technologien bewertet und Funktionen explizit entschieden. Dadurch entsteht ein gemeinsames, verifizierbares Verständnis: „Ja, das ist die Maschine, die wir bauen wollen.“



### Formale Verhaltensmodellierung

Das entschiedene Verhalten wird in ein formales, zustandsbasiertes Modell überführt. Jeder Zustand, jede Aktion und jeder Übergang wird sichtbar und überprüfbar – wie ein elektrischer Schaltplan für Verhalten. Es gibt keine versteckten Annahmen. Keine undefinierten Zustände.



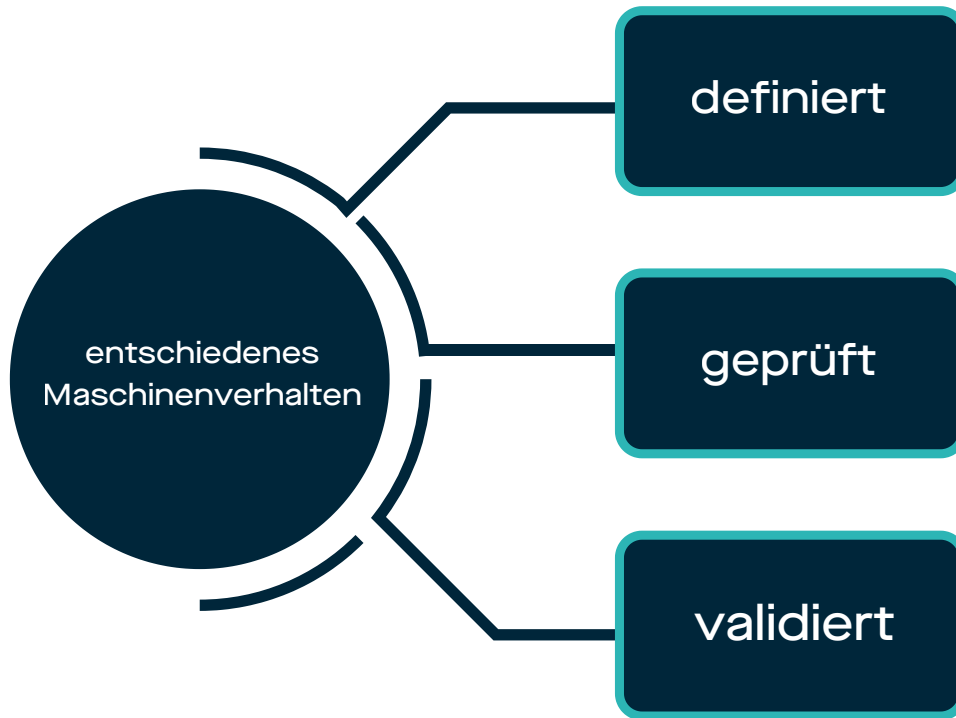
### Deterministische Implementierungsabbildung

Aus dem validierten Modell wird die Ausführungslogik strukturell abgeleitet. Die Implementierung folgt der Entscheidung, nicht der Interpretation. Die resultierende Steuerungslogik ist konsistent, standardisiert und lange vor der Inbetriebnahme bereit.

Die Implementierung wird zur Ausführung einer Entscheidung, nicht zu dem Ort, an dem die Entscheidung erst getroffen wird.

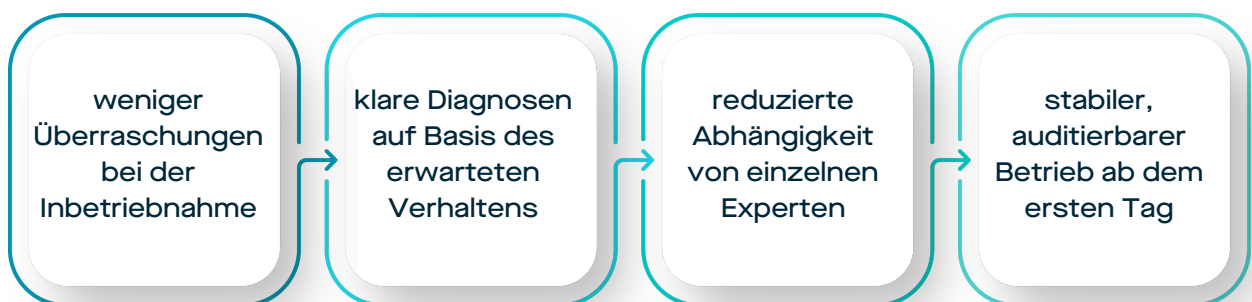
# Wenn Verhalten entschieden ist, wird Integration vorhersagbar.

Integration und Inbetriebnahme sind nicht deshalb herausfordernd, weil Maschinen komplex sind, sondern weil das Maschinenverhalten erst dort sichtbar wird. Mit Selmo ändert sich das grundlegend.



Integration wird zum kontinuierlichen Abgleich zwischen Realität und einer bekannten Struktur. Virtuelle Validierung bestätigt die Korrektheit frühzeitig. Die Inbetriebnahme vor Ort konzentriert sich auf Abgleich, nicht auf Interpretation. Das Ergebnis ist nicht nur Geschwindigkeit, sondern Klarheit. Ingenieure arbeiten mit einem transparenten Modell. Bediener erhalten kontinuierliche Einsicht in Maschinenzustand und -verhalten.

## Betriebliche Auswirkungen



Inbetriebnahme ist keine Suche nach Verhalten, sondern dessen Bestätigung.

# Mit Klarheit entscheiden, bevor Sie sich festlegen.

Die Arbeit mit Selmo ist bewusst so strukturiert, dass Unsicherheit bereits aus der Entscheidung selbst entfernt wird.

Bevor die Implementierung beginnt, schaffen wir ein gemeinsames Verständnis darüber, wie sich die Maschine verhalten soll und was sie nicht tun darf.

Damit wird sichergestellt, dass Entscheidungen bewusst, transparent und ohne Druck getroffen werden.

## Klare Schritte zu hochwertigen, sicheren und intelligenten Maschinen

### 01

#### Strategische Klärung:

Der erste Schritt ist eine fokussierte Strategiesitzung. Dabei geht es nicht um Verkauf, sondern um Klarheit:

- wo Maschinenverhalten derzeit implizit ist
- wo Entscheidungen ohne Transparenz getroffen werden
- und wo strukturelles Risiko besteht

Auf dieser Basis prüfen wir gemeinsam, ob Selmo zu Ihrem Kontext passt.

### 02

#### Potenzialanalyse & PTF:

Im nächsten Schritt, wird das beabsichtigte Maschinenverhalten formal mit dem PTF-Framework analysiert.

Dabei werden Prozess, Technologie und Funktion auf eine gemeinsame, klare Referenz ausgerichtet.

Das Ergebnis ist ein klarer Entscheidungszeitpunkt: Das Maschinenverhalten wird bewusst und eindeutig festgelegt.

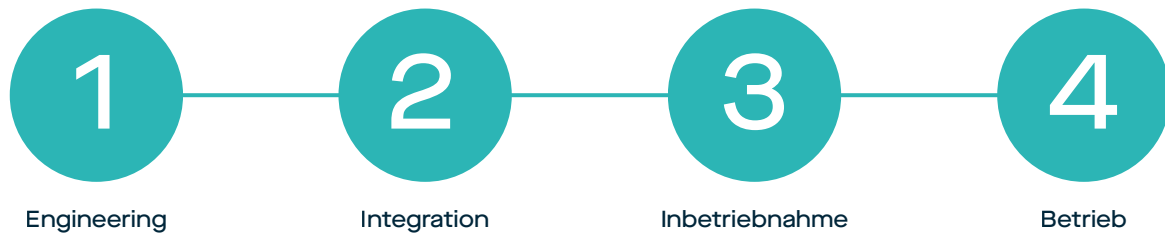
„Ja – das ist das Verhalten, für das wir stehen wollen.“

Kein Verhalten wird implementiert, bevor bewusst darüber entschieden wurde.



# Von definiertem und entschiedenem Verhalten zu kontrollierter Ausführung.

Sobald über das Maschinenverhalten formal entschieden wurde, folgt die Implementierung einem klaren und kontrollierten Pfad. Das Verhaltensmodell wird zur einzigen maßgeblichen Referenz für alle nachfolgenden Schritte:

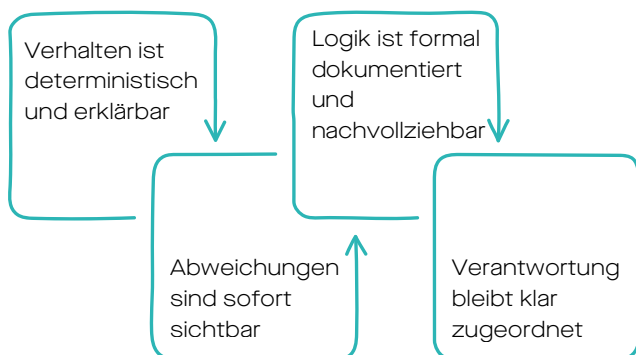


Die Ausführungslogik wird aus dem Modell abgeleitet, nicht aus Anforderungen interpretiert. Das Ergebnis:



Es entsteht nicht nur eine laufende Maschine, sondern ein kontrolliertes System.

## Betriebliches Ergebnis



Die Maschine geht als Premium-Asset in Betrieb. Das schafft Vertrauen, nicht nur beim Anlauf, sondern über den gesamten Lebenszyklus hinweg.

Eine Selmo-basierte Maschine verlässt sich nicht auf Hoffnung. Sie stützt sich auf eine entschiedene Struktur.

# Warum KI strukturierte Entscheidungsgrundlagen braucht

Künstliche Intelligenz scheitert nicht an fehlenden Algorithmen. Sie scheitert an fehlender Struktur. KI-Systeme erfordern:

- eindeutige Zustände
- klar definierte Übergänge
- zuverlässige, kontextualisierte Daten

Ohne formal entschiedenes Maschinenverhalten verlieren Daten ihre Aussagekraft, Ereignisse werden mehrdeutig und Ergebnisse nicht erklärbar. In solchen Umgebungen können KI-Entscheidungen nicht validiert werden und sind daher nicht verantwortungsvoll einsetzbar.

## Die Selmo-Basis

Indem Maschinenverhalten in eine formale, deterministische Struktur überführt wird, liefert Selmo das, was KI grundlegend benötigt:

- klaren operativen Kontext
- erklärbare Systemzustände
- nachvollziehbare Ursache-Wirkungs-Zusammenhänge

KI optimiert auf Basis klarer Vorgaben, sie entscheidet nicht im Blindflug.



KI ist nur so vertrauenswürdig wie die Struktur,  
auf der sie arbeitet.



# Über Maschinenverhalten zu entscheiden ist nicht länger optional.

Industrielle Organisationen erreichen einen Punkt, an dem implizites Maschinenverhalten nicht mehr ausreicht.

Die Komplexität nimmt zu. Verantwortung wird klarer. Die Erwartungen an Transparenz, Sicherheit und Erklärbarkeit steigen weiter.

In diesem Umfeld reicht es nicht mehr, reine Annahmen über das Maschinenverhalten zu tätigen. Es erfordert:



explizite  
Entscheidung



formale Definition



kontinuierliche  
Verifikation

## Selmo ist für Organisationen, die Klarheit statt Annahmen und Struktur statt Interpretation wählen.



### Positionierung

Selmo ist nicht für jeden. Sondern für die, die es ernst meinen.

Es ist für Organisationen, die:

- verstehen wollen, was ihre Maschinen tatsächlich tun
- Verantwortung für dieses Verhalten übernehmen
- und diese Klarheit über Teams, Lieferanten und Jahre hinweg bewahren wollen

Klarheit ist kein Feature.  
Sie ist eine Entscheidung.

## Sind Sie bereit, **Performance-Steigerung** strukturell zu verankern?

6 klar messbare Effekte, die durch entschiedenes Maschinenverhalten entstehen

### **Bis zu 40 % kürzere Projektlaufzeiten.**

Wenn Maschinenverhalten vor der Programmierung entschieden wird, entfallen späte Iterationen und Interpretationsschleifen. Projekte werden planbar. Änderungen bleiben kontrollierbar. Time-to-Market wird messbar verkürzt.

### **OEE von 85–95 % nachhaltig erreichbar.**

Höhere Verfügbarkeit, reproduzierbare Performance und weniger Störungen machen OEE kontrollierbar. Nicht geschätzt. Nicht erklärt. Sondern strukturell abgesichert.

### **Bis zu 70 % weniger ungeplante Stillstände.**

Deterministisch definiertes Maschinenverhalten verhindert strukturelle Abweichungen. Fehler werden sofort sichtbar. Diagnosen basieren auf Referenz – nicht auf Interpretation.

### **Bis zu 50 % schnellere Inbetriebnahme.**

Durch vorab validierte, deterministische Verhaltensmodelle wird Unsicherheit in der Inbetriebnahme strukturell reduziert. Abnahmen werden klarer, schneller und reproduzierbar – nicht erfahrungsabhängig.

### **Bis zu 20 % höherer Output.**

Stabile, deterministische Prozesse eliminieren wiederkehrende Verluste und Engpässe. Leistungssteigerung entsteht nicht durch Druck – sondern durch Struktur.

### **Bis zu 50 % geringerer Wartungsaufwand.**

Transparente Zustände und eindeutige Logik reduzieren die Zeit für Fehlersuche signifikant. Abweichungen werden unmittelbar sichtbar. Wartung wird strukturiert, gezielt und datenbasiert – statt reaktiv.

## Wenn Sie für Premium-Leistung bezahlen, sollten Ihre Maschinen nicht auch Premium-Verhalten liefern?

**Treffen Sie eine strukturelle Entscheidung.  
Jetzt Strategiegespräch vereinbaren.**

Selmo Technology GmbH  
einfach@selmo.at  
+43 3136 20755

