BOTFORCE

Practical guide

From script to goal.

Agentic AI in the enterprise: how AI agents take on real work – and what keeps them in check

BOTFORCE GmbH, Vienna

With a continuous case story, seen through the lens of information systems

As of

09/2026

6 CHAPTERS

botforce-technology.com

ABOUT THIS TEXT

Why a guide built on a case story – and how it works

AI agents are usually written about in one of two registers: as a promise that software will soon replace entire departments, or as a warning about systems nobody controls any more. This text chooses a third register, that of information systems. It does not start by asking what a technology can do, but which task it takes on in a company, how it interacts with people, processes and existing systems, and how you can tell whether it pays off.

The text follows the six blocks of the **Agentic AI cheat sheet**. What fits into a few lines there is explained in depth here. Every chapter opens with a **scene** from a continuous case story. Then comes the subject itself, placed in context in a box headed **The information systems view**. Each chapter ends with a **key takeaway** and **questions for your organisation**; notes on them are in the appendix.

The case story

Aurelia Haustechnik GmbH is a wholesaler of plumbing and heating equipment based in St. Pölten, Austria, with 420 employees, just over 900 suppliers and around 4,000 incoming invoices a month. The story follows:

Miriam Leitner	heads accounting and becomes owner of the first agent.
Jonas Hofer	is an information systems specialist in IT, responsible for processes and automation.
Sabine Kraus	has processed supplier invoices for eleven years and knows every difficult supplier.
Karl Weninger	is the commercial managing director and has to make the final call.

The company, people, suppliers and figures in the story are entirely fictional. They illustrate relationships and are neither measurements nor references. Statements about the EU AI Act reflect the state after the Digital Omnibus (September 2026) and are not legal advice.

Contents

PROLOGUE

Monday, 7:40 a.m.

When Miriam Leitner switches on her computer one Monday in February, 312 invoices have arrived over the weekend. It is not a bad Monday. It is an ordinary one.

Aurelia buys goods from just over 900 suppliers, and each of them sends invoices in its own way: as a PDF attached to an email, through a supplier portal, occasionally still on paper. Since 2019 a software robot has collected the invoices from the portals every night. Since 2021 a document recognition system has read out supplier, amount and purchase order number. Since then, a good six out of ten invoices post themselves without anyone touching them. The rest land with Miriam's team – around 1,500 a month.

"We automated the easy cases long ago," Miriam says when asked. "What's left are the ones where you have to think." A missing purchase order number. A supplier who has changed its bank details. A partial delivery that matches the order but not the delivery note. These cases cost time, they cost early-payment discounts, and they cost the patience of people who really have other things to do.

That Monday, shortly after eight, Jonas Hofer is standing in her doorway. He has a coffee in his hand and a question on his mind that he has been turning over for weeks: "What if we stopped just passing the difficult cases on and had them handled instead?"

Miriam laughs. "By whom? Your robot?"

"No," says Jonas. "By something that understands a goal."

This text tells what happens over the following twelve months – and explains, at every stage, what lies behind it.

CHAPTER 01

What an agent is

IN THIS CHAPTER

- Distinguish an AI agent precisely from a chatbot and a copilot.
- Explain the loop of perceiving, reasoning, planning, acting and verifying using a business transaction.
- Understand why delegating goals is an organisational question, not just a technical one.

SCENE · THE INVOICE WITHOUT A PO NUMBER

Jonas puts an invoice on Miriam's desk that the team had to leave on Friday: Kranzl Armaturen, €2,380, no purchase order number. Sabine Kraus explains how she would solve it. Look up the supplier number in the ERP. Check Kranzl's open orders – there are two. Compare amount and delivery date. Match the right order, post the invoice, leave a note saying why. Ten minutes, if nobody interrupts.

"That's six steps," says Jonas, "and none of them is written down like that in a work instruction. The work instruction says: check the invoice and post it. Sabine knows the goal and finds the way herself."

"Our robot knows a way too," says Miriam.

"Exactly one," says Jonas. "The one we wrote down for it."

A goal instead of a sequence of steps

An **AI agent** is a software system that is given a goal instead of a sequence of steps and pursues that goal in a loop. Classic automation works procedurally: someone describes every step in advance, and the software executes it. An agent works towards a goal: it receives a desired end state – "the invoice is posted correctly and documented" – and determines the steps to get there for each case itself, within the limits it has been given.

This has only become possible since large language models learned to read documents, recognise intentions and formulate intermediate steps (Chapter 3). The model on its own is not yet an agent, though. It becomes one through tools with which it acts in real systems, and through the loop in which it verifies its own result.

The loop

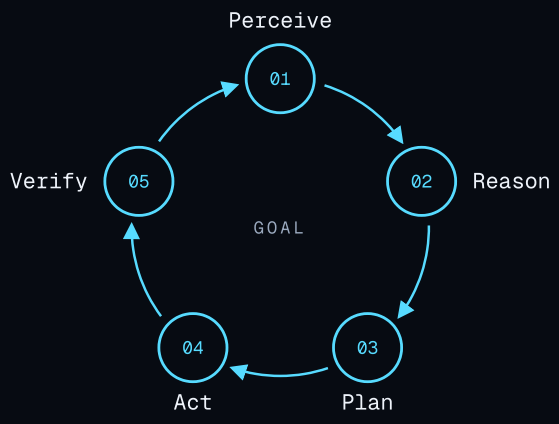

FIG. 1 - THE AGENT LOOP: FIVE STEPS AROUND A GOAL.

The five steps describe how any goal-directed clerical work is done. Using the case from the scene:

01 **Perceive**	The agent reads the invoice and notices that the purchase order number is missing.
02 **Reason**	The supplier number leads to the supplier's open orders.
03 **Plan**	It queries the open orders in the ERP and decides how to recognise the right one: amount, quantity, delivery date.
04 **Act**	It matches the right order and prepares the posting – or carries it out, if it is allowed to.
05 **Verify**	It checks the result against the delivery note and records why it chose this order.

If the result does not fit, the loop starts again. It ends when the goal is reached or when a rule makes the agent stop – for instance because it finds two equally good candidates and is not supposed to guess. The case then goes to a person, together with the agent's reasoning.

Chatbot, copilot, agent

In everyday use, three terms are often mixed up. They describe different degrees of independence:

	WHAT IT DOES	EXAMPLE AT AURELIA	WHAT IT ADDITIONALLY NEEDS
Chatbot	answers questions, does not act	"Which discount period applies to Kranzl?"	knowledge and sources
Copilot	prepares, a person executes	marks the matching order, Sabine posts	read access to data
Agent	executes itself and reports back with a result or an exception	matches, posts and documents its choice	its own identity, limits, a record

TABLE 1 - THREE DEGREES OF INDEPENDENCE.

Two questions help. **Who sets the steps?** If the system chooses its path and tools itself, it is an agent. **Who presses "execute"?** As long as a person confirms every action, that person is responsible for every step; as soon as the system acts on its own, the organisation has to secure that responsibility in a different way. That is

what Chapters 4 and 5 are about.

THE INFORMATION SYSTEMS VIEW

The agent as a delegate

The word agent does not come from computer science. In business economics, **principal–agent theory** describes relationships in which a principal delegates a task to an agent. The agent knows more about its own work than the principal, and its goals do not automatically match the principal's. The theory has remedies for this: clear contracts, controls, reporting duties.

Both problems return with software agents. Nobody can observe every intermediate step, and a language model optimises for plausible answers, not for the company's goals. Here the remedies are called policy, audit trail and human oversight. Anyone introducing an agent is therefore designing not just a technical system but a **delegation relationship** – an idea that information systems, with its view of people, technology and organisation as one whole, is well placed to grasp.

KEY TAKEAWAY

An agent is given a goal, not a procedure. That shifts responsibility – and the organisation has to arrange it anew.

QUESTIONS FOR YOUR ORGANISATION

1.1 Distinguish chatbot, copilot and agent using the question "Who executes?" and give an example from accounting for each.

1.2 Describe the five steps of the agent loop using a task from your own work.

1.3 Which problems of principal–agent theory arise with software agents, and which mechanisms address them?

CHAPTER 02

Script, RPA or agent?

IN THIS CHAPTER

- Distinguish scripts and interfaces, RPA and agents as forms of automation.
- Name the process characteristics that determine the right technology.
- Explain why process mining and process metrics come before any automation decision.

SCENE · THE BOT THAT STOPPED

Jonas remembers the October before well. Overnight, the largest supplier portal had redesigned its login form. The robot that collected invoices there every night could no longer find the password field and aborted. It reported the error correctly – to a mailbox nobody read. For three nights. Around 400 invoices reached accounting late, and for 23 of them the discount period had expired.

“I thought this was automated,” Karl Weninger had said at the time.

“It is,” Jonas had replied. “Just not the surprise.”

Now, in February, Karl wants to know whether an agent would be the same surprise, only more expensive.

A spectrum, not a choice between old and new

Automation is not a single technology but a spectrum with three areas.

Scripts and interfaces connect systems through defined channels (APIs). They are fast, cheap and robust – provided the systems involved offer such channels and the rules can be described completely.

Robotic process automation (RPA) comes in where interfaces are missing. Software robots operate applications through their user interface, the way a person would: they click, type and read screen content. That makes RPA quick to introduce. It also makes it sensitive to every change in a screen and blind to every case the script does not anticipate.

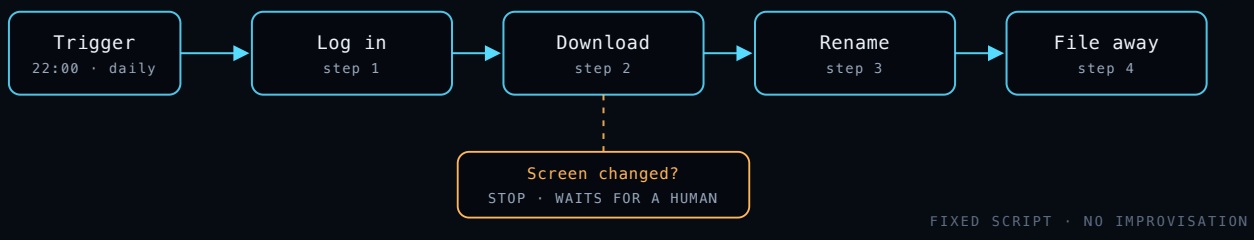

FIG. 2 – AN RPA BOT REPLAYS ITS SCRIPT; THE UNEXPECTED SCREEN STOPS IT.

Agents pursue goals instead of scripts. They cope with variants, unstructured documents and incomplete information. In return, they cost more to run, work probabilistically and need governance of their own.

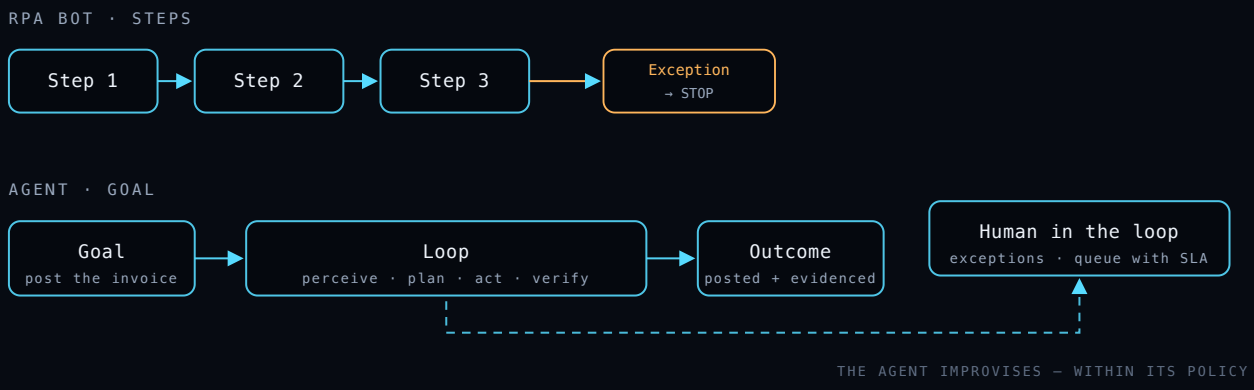

FIG. 3 - SCRIPT VERSUS GOAL: WHERE THE BOT STOPS, THE AGENT ESCALATES IN A CONTROLLED WAY.

Which technology fits?

The choice follows from the characteristics of the process, not from enthusiasm for a technology:

CHARACTERISTIC	SCRIPT / API	RPA	AGENT
Interfaces	available	missing	available or reachable through tools
Rules	fixed and fully describable	fixed	partial; case-by-case judgement needed
Variants	few	few, stable user interface	many
Data	structured	structured, on screen	also unstructured: emails, PDFs, free text
Exceptions	rare	rare	frequent, but with recognisable patterns
Running cost per case	very low	low	higher: model calls, oversight
Typical risk	interface changes	user interface changes	wrong judgement without control

TABLE 2 - PROCESS CHARACTERISTICS AND MATCHING AUTOMATION.

In practice, the forms do not exclude each other; they complement each other. At Aurelia, the handover to payments remains a script. The download from two portals without an interface remains RPA – this time with monitoring that someone actually reads. The agent takes on the cases where both fail, and even calls the RPA robot as a tool when needed.

When not to use an agent

Three warning signs speak against an agent, at least for now:

- **No real cases to test against.** Without historical cases with known correct outcomes, quality cannot be measured (Chapter 3).
- **Nobody owns the exceptions.** An agent hands cases back to people. If they sit there unread, nothing is gained – the October mailbox sends its regards.
- **The process is rare.** Where ten cases arise per month, neither building nor supervising an agent pays off.

THE INFORMATION SYSTEMS VIEW

Measure first, then automate

Information systems assesses automation through **process metrics**. For invoice processing, these are above all lead time, cost per case and the **straight-through processing rate** (STP): the share of cases that pass through without a human touch. At Aurelia it is 62 per cent. Every stage of automation is an attempt to raise this rate without increasing the risk of errors and loss of control.

How a process actually runs is shown by **process mining**: it reconstructs the flow from the event logs of the systems – timestamps, status changes, handlers. When Jonas analyses a year of invoice data, he finds that 71 per cent of the manual cases go back to five exception patterns: missing purchase order number, price deviation, partial delivery, changed master data, illegible documents. The exceptions are varied, but not arbitrary. That is the real foundation of the business case.

WORKED EXAMPLE · ASSUMPTIONS OF THE CASE STORY

1,500 manual invoices a month × 9 minutes = 225 working hours. If an agent completes 60 per cent of these cases, around 135 hours a month are freed up. Against this stand model calls, operations, evals, the remaining oversight and the introduction itself. The more important question is asked by Karl Weninger: “What do we use the time for?”

KEY TAKEAWAY

Technology follows the process: interfaces where possible; RPA where nothing else works; an agent where judgement is needed – and only where someone owns the exceptions.

QUESTIONS FOR YOUR ORGANISATION

2.1 Why is RPA quick to introduce but fragile in operation?

2.2 Using Table 2, assign three processes from your company to a form of automation and justify your choice.

2.3 What role do process mining and the straight-through processing rate play in the business case for an agent?

CHAPTER 03

The building blocks

IN THIS CHAPTER

- Describe the four building blocks: model, context, tools and evaluation.
- Explain why language models guarantee no truth – not even with reasoning – and how retrieval-augmented generation deals with that.
- Understand the tool layer as an agent's most important control surface.

SCENE · THE WHITEBOARD

Karl Weninger wants to understand what he is approving before he approves it. Jonas draws four boxes on the whiteboard in the meeting room.

"This is the model. It understands language, but it knows nothing about Aurelia. This is where the context goes in: our contracts, orders, work instructions. These are the hands – tools the agent may use to look things up in the ERP and post, and nothing else." He taps the fourth box. "And this is why I can sleep at night. We measure before we entrust it with anything."

Karl looks at the boxes for a while. "And which of them makes mistakes?"

"The first one," says Jonas. "The other three are there so that we notice."

The model: understanding without guarantees

At the core is a **large language model** (LLM). It was trained on vast amounts of text to predict the next word, and in doing so learned to understand language, recall knowledge and follow instructions. Today's leading models are then also trained with reinforcement learning to think tasks through in intermediate steps – which makes them far stronger at multi-step tasks. Just as important is what a language model is not: not a database, not a calculator and not a source of guarantees. Where knowledge is missing, a plausible guess often comes more naturally to the model than "I don't know".

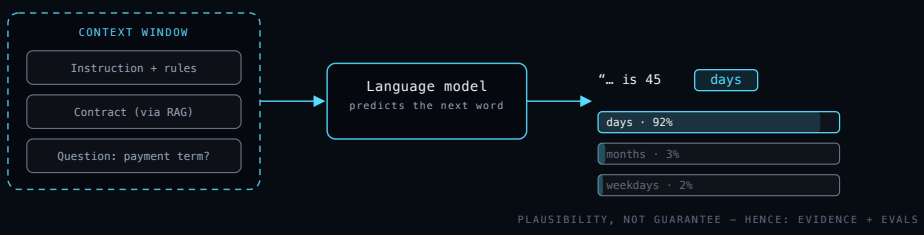

FIG. 4 – AN LLM PRODUCES ITS ANSWER WORD BY WORD FROM PROBABILITIES OVER ITS CONTEXT.

Context and RAG: the company's facts

Everything the model gets to see for a task – instruction, rules, documents – sits in its **context window**. What is not there does not exist for the model at that moment. Windows now often hold a million tokens or more, but less of that is used reliably – which is why the discipline is called **context engineering**: show the right

things, not as much as possible. Ask a model without access to the supply contract about the payment term, and it may answer “30 days”, because that term appears frequently in its training data. Such a confidently stated falsehood is called a **hallucination**.

Retrieval-augmented generation (RAG) counters this: before every answer, the system retrieves the relevant passages from the company’s own documents and places them in the model’s context. The model then quotes the real clause: 45 days, section 7.2. A rule of thumb for companies: improve instructions and RAG first; retrain a model on your own data (fine-tuning) only when it is demonstrably necessary.

Tools: the agent’s hands

A model on its own can only produce text. **Tools** give it defined capabilities: query an order in the ERP, propose a posting, start an RPA robot, send an email to a supplier. With **function calling**, each tool is described with a name, purpose and parameters; the model decides when to call which one with which values. The **Model Context Protocol** (MCP), vendor-neutral under the Linux Foundation since December 2025, unifies how systems offer their tools; with A2A, agents hand tasks to each other.

For governance, this layer is central. What is not released as a tool, the agent cannot do, and a limit a tool enforces technically, it cannot argue its way around. Care is needed with general-purpose tools such as screen or browser control (“computer use”) and code execution: they open everything the agent’s account may do.

Evals: quality as a measurement

Because language models work probabilistically, **evaluation** (evals) takes the place of classic software testing: a collection of real cases with known correct outcomes against which the system is checked automatically after every change – after every new instruction, every model switch, before every release. With agents, the path counts too – which tools were called with which values – and every case runs several times, because the same case can turn out differently.

From the previous year’s invoices, Jonas puts together such a **golden set** of 1,200 cases, deliberately including all five exception patterns. The first version of the agent handles 96.4 per cent of them correctly. That sounds good until you look at the 43 errors one by one: in 19 of them it would have matched a partial delivery incorrectly. The list becomes the work plan for the following weeks.

THE INFORMATION SYSTEMS VIEW

An agent as a layered architecture

Information systems likes to describe business information systems in layers. That helps here too:

- **Knowledge layer** – context, RAG, master data: what the agent judges on.
- **Processing layer** – model and planning: how it judges.
- **Integration layer** – tools, interfaces, RPA: what it can bring about.
- **Control layer** – evals, policy, record, oversight: whether it can be trusted.

The separation has a tangible benefit. Language models are replaced at short intervals. If knowledge, tools and controls are cleanly separated from the model, the model can be swapped, and the same evals prove that its successor works at least as well. The model becomes a replaceable component; the company’s knowledge stays in-house.

KEY TAKEAWAY

The model supplies understanding, the context the facts, the tools the effect and the evals the proof. If one of the four is missing, the agent lacks something essential.

QUESTIONS FOR YOUR ORGANISATION

3.1 Why do language models hallucinate, and how does RAG reduce this risk?

3.2 Why is the tool layer considered an agent's most important control surface?

3.3 How would you put together a golden set for an agent in your own department?

CHAPTER 04

The governance ring

IN THIS CHAPTER

- Explain the four controls: identity, policy, audit and oversight.
- Justify why rules must be enforced technically rather than only written into the prompt.
- Understand human-in-the-loop as process design, not as blanket control.

SCENE · €12,400

Tuesday, 2:02 p.m., week three of the pilot. An invoice from Vettori Pumpen arrives: €12,400. The agent recognises supplier, order and delivery note; everything matches. It could post the invoice.

It does not. A second later the case is in Sabine Kraus's work queue – with the matched order, the proposed posting and one sentence of reasoning: "Amount above the €10,000 limit for autonomous postings."

After her meeting, Sabine opens the case, looks at the delivery note and approves it. The agent posts. The log shows four lines. Just under 30 minutes lie between the first and the last, and nobody had to call anybody.

"In the past," Sabine later tells Miriam, "I would have had to find the invoice first, then check it, then post it. Now I only check. That's the part I'm actually here for."

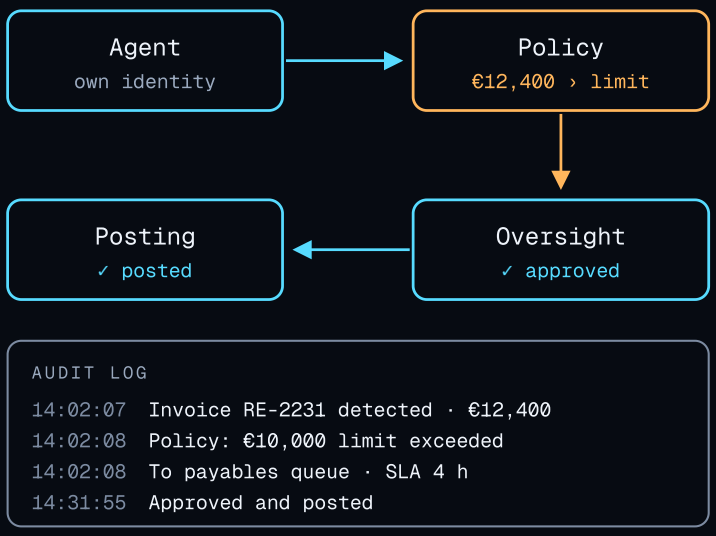

FIG. 5 - THE POLICY APPLIES, A PERSON APPROVES, EVERY STEP IS IN THE AUDIT LOG.

The capacity to act needs limits

An agent that acts in real systems is as valuable as it is risky. Enterprise-ready agents therefore sit inside a ring of four controls that complement one another.

Identity

Every agent gets its own identity in the company's identity system – vendors now treat agents as a distinct identity type – with an accountable owner, never a shared service account. Its permissions are tightly scoped: the accounts payable agent may read supplier invoices and create postings, but may not change supplier master data. Its credentials are short-lived and expire after minutes rather than months. That way it is always traceable who acted, and access reviews work just as they do for human employees.

Policy

The agent's limits are laid down as rules: postings up to €10,000 autonomous, four-eyes approval above that; changes to bank details never autonomous; access to accounts payable transactions only. What matters is **where** these rules are enforced. A rule in the prompt is a request to a probabilistic model that it can misunderstand. A rule in the tool layer is a fact: the posting tool refuses, however convincingly the agent argues. Policies are changed like software – with review and version history.

This matters most because of **prompt injection**: instructions hidden in emails, PDFs or web pages the agent reads. Language models cannot reliably separate instructions from data, and security agencies consider the problem not fully solvable. What counts, therefore, is what a deceived agent can do – and that is set by tools and policy, not by the prompt.

Audit

Every input including third-party content, every decision with its reasoning, every tool call and every result is recorded tamper-evidently, ideally hash-chained so that later changes stand out. The reasoning alone is not enough – what a model gives as its reason is not always the real reason. The audit trail answers the question that auditors, internal audit and management will ask: *why did the system do that?* At the same time it is the best tool for root-cause analysis, because every case can be traced step by step.

Oversight

Human-in-the-loop does not mean a person looks at everything. Designed well, exceptions land in a queue with a service level (SLA), together with the case, the reasoning and the agent's proposal. People decide the cases that need judgement instead of retyping routine. This includes a stop switch for each agent that works outside the model – disabling the identity, revoking tokens – and that the business team may operate itself, not IT after a ticket.

THE INFORMATION SYSTEMS VIEW

An internal control system for a new kind of actor

For accounting and information systems, the governance ring is nothing alien. It applies proven principles of the **internal control system** (ICS) to a new kind of actor:

- **Segregation of duties:** whoever posts does not change master data.
- **Four-eyes principle:** mandatory above defined thresholds.
- **Complete records:** every transaction is fully documented.
- **Access reviews:** regular, including for agents.

What is new is the pace. In an hour, an agent can trigger as many transactions as a team does in a week. Controls that are sufficient as spot checks for people must therefore work completely and automatically for agents.

KEY TAKEAWAY

Trust in an agent does not come from its intelligence but from its limits: its own identity, rules in the tool layer, a complete record and people who decide the exceptions.

QUESTIONS FOR YOUR ORGANISATION

4.1 Why is it not enough to state a posting limit in the prompt?

4.2 Using the scene, explain how the four controls work together.

4.3 Which principles of the internal control system transfer directly to agents, and where do they need adapting?

CHAPTER 05

Autonomy in levels

IN THIS CHAPTER

- Distinguish the levels of autonomy L0 to L4.
- Name the metrics that justify moving up a level.
- Describe the organisational prerequisites for operating agents.

SCENE · THE FIRST EIGHT WEEKS

For the pilot, Karl Weninger set one condition: “At the start, the agent posts nothing that a person hasn’t seen.” That suits Jonas, and Miriam too.

For eight weeks, the agent therefore only proposes. Sabine and two colleagues confirm every posting with a click or correct it. Every correction goes into the golden set as a new case. In week three, the proposals match the people’s decisions in 94 per cent of cases; in week eight, in 98.1 per cent. Exceptions in the queue are handled after two and a half hours on average; the agreed service level is four.

Then Miriam, Jonas and Karl sit down and decide: from now on, the agent posts invoices up to €10,000 with a clearly matched order on its own. Invoices with price deviations it continues only to propose. “We’re not switching on autonomy,” says Miriam. “We’re releasing a group of cases.”

A staircase, not a threshold

Autonomy is not a switch that is flipped once. It grows in levels:

L0	**Manual work**	People do every step.
L1	**Scripted automation**	RPA and scripts carry out fixed procedures without judging.
L2	**Intelligent automation**	Machine learning inside the process, for instance to read documents.
L3	**Assisted agents**	The agent proposes, a person confirms every action.
L4	**Governed autonomy**	The agent acts within limits, people decide the exceptions.

TABLE 3 – LEVELS OF AUTONOMY L0 TO L4.

Moving up is not a technology exercise but a matter of building trust. It happens per process and often per group of cases: at Aurelia, the agent works at L4 for unambiguous invoices up to €10,000 and remains at L3 for invoices with price deviations. It moves up a level only when evals, audit trail and exception rate justify it – not because a project plan says so. The levels are a maturity model for processes, not a standard.

How to tell the time is right

- **Agreement rate:** how often does the agent decide the way people do while operating at L3?
- **Eval result:** stable on the golden set, with no regression against the previous version.
- **Exception rate and lead time:** can the team handle the queue within the service level?
- **Errors after posting:** reversals or corrections per 1,000 cases, compared with the manual process.

Operations need roles

An agent without anyone responsible is an incident waiting to happen. Every agent therefore has a named **owner** in the business team who answers for its results – at Aurelia, that is Miriam. Capacity for human-in-the-loop is planned like any other work. Error cases are reviewed at a fixed rhythm and added to the golden set. And the team rehearses the switch to a new model before a vendor forces it.

THE INFORMATION SYSTEMS VIEW

Maturity is only as high as the weakest dimension

Maturity models are a familiar tool in information systems for describing how organisations develop. For operating agents, five dimensions count: **evals**, **governance**, **operations**, **roles** and **continuous improvement**. An organisation is only as mature as its weakest dimension: the best agent is of little use if nobody handles the exceptions.

Work design matters just as much. Clerical work shifts from entering data to making decisions. Many find that more attractive, but it requires different skills and has to be supported – through training, clear responsibilities and honest communication about what changes for whom.

KEY TAKEAWAY

Autonomy is earned, not switched on: level by level, per process and group of cases, backed by evals, a record and a manageable exception rate.

QUESTIONS FOR YOUR ORGANISATION

5.1 How do L3 and L4 differ, and what evidence does the move require?

5.2 Why can one and the same process run at two levels at the same time?

5.3 Which roles does operating an agent require, and what happens if one of them is missing?

CHAPTER 06

EU AI Act: where agents sit

IN THIS CHAPTER

- Explain the risk-based approach of the EU AI Act and its four tiers.
- Distinguish the roles of provider and deployer.
- Know the key dates after the Digital Omnibus.

SCENE · "DOES THIS AFFECT US?"

In April, Karl Weninger reads a newspaper article about AI regulation and forwards it to Miriam and Jonas with a single line: "Does this affect us?"

Jonas does not reply with an assessment but with a document he keeps anyway: the **agent profile**. It sets out what the accounts payable agent is for, which data it processes, which decisions it makes, whom those decisions affect, which limits apply and who supervises it. He adds: "The legal assessment has to come from the law firm. But without these facts, they can't do it."

A week later, the profile is with the law firm Aurelia works with.

Risk, not technology

The **EU AI Act** (Regulation (EU) 2024/1689) is the first comprehensive AI law. It applies directly in all member states and covers everyone who provides or deploys AI systems in the EU, including providers from outside the EU. Its core is a risk-based approach: the greater a system's risk to people, the stricter the obligations.

FIG. 6 - THE EU AI ACT RISK PYRAMID: THE HIGHER THE TIER, THE STRICTER THE OBLIGATIONS.

- **Prohibited practices** (Art. 5): for example social scoring or manipulative systems. The Digital Omnibus added AI systems that generate non-consensual intimate imagery or child sexual abuse material; that prohibition applies from 2 December 2026.
- **High-risk** (Annexes I and III): AI in areas such as recruitment, credit decisions, critical infrastructure or regulated products. Permitted, but with strict obligations on risk management, data quality, documentation and human oversight (Art. 14, for deployers Art. 26), applicable from 2 December 2027 and 2 August 2028 respectively.

- **Transparency obligations** (Art. 50): providers make sure people can tell they are interacting with an AI system and mark AI-generated content in machine-readable form; deployers disclose deepfakes, among other things. The Commission published guidelines on 20 July 2026.
- **Minimal risk:** the large remainder, with no special requirements.

Classification depends on the **intended purpose**, not on the technology. An agent that matches invoices and proposes postings typically sits in the lower tiers; if it answers supplier queries, transparency obligations are added. The same agent, used to pre-screen job applications, would be a high-risk system under Annex III.

Roles: provider and deployer

The AI Act assigns obligations by role. The **provider** is whoever develops an AI system and places it on the market. The **deployer** is whoever uses it under their own responsibility – at Aurelia, the company itself. Important in practice: substantially modifying a system or using it under your own name can shift you into the provider role, with its obligations. In projects with service providers, the question of roles therefore belongs in the contract: who documents, who monitors, who reports incidents?

Two further points affect almost every deployer. Since 2 August 2025, the providers of large foundation models (general-purpose AI) have had obligations of their own; the model provider's documentation belongs in the file on your own AI system. And since 2 February 2025, providers and deployers have had to take measures to support **AI literacy** among their staff (Art. 4), appropriate to their role and the systems in use; since the Digital Omnibus, they need not guarantee a particular level. At its core little changes: whoever works the queue must understand what the agent can do, where it errs and when to stop it.

The timeline

DATE	WHAT APPLIES
2 February 2025	Prohibitions (Art. 5) and AI literacy (Art. 4)
2 August 2025	Obligations for providers of general-purpose AI models
2 August 2026	Transparency obligations under Art. 50
2 December 2026	Deadline for machine-readable marking of systems placed on the market before August 2026 (providers); new prohibition on AI-generated intimate imagery
2 December 2027	High-risk obligations under Annex III
2 August 2028	High-risk obligations for AI in regulated products (Annex I)

TABLE 4 - TIMELINE AFTER THE DIGITAL OMNIBUS, IN FORCE SINCE 27 JULY 2026.

Violations of the prohibitions can draw fines of up to €35 million or 7 per cent of worldwide annual turnover, and up to €15 million or 3 per cent for most other obligations; for SMEs the lower figure applies. Enforcement is national: in Germany, the Federal Network Agency is responsible under the AI Market Surveillance Act of July 2026; in Austria, no market surveillance authority has been designated so far – RTR's AI service desk informs and advises. For foundation models, the AI Office is responsible EU-wide. Realistically, supervision starts with questions – and the best answer is a well-kept file.

THE INFORMATION SYSTEMS VIEW

Compliance starts in the architecture

It is striking how many requirements of the AI Act have a technical counterpart in the governance ring. Human oversight corresponds to the queue with a stop switch, documentation duties to the audit trail, transparency to a marking function in the platform, the inventory to the agent profile. Building these functions in from the start creates the factual basis that a legal assessment needs.

A sensible order in the company: first an **inventory** of all AI systems with profiles, then the **classification** by lawyers, then the **marking**, and finally – if needed – the documentation for high-risk systems.

NOT LEGAL ADVICE

This chapter gives an overview as of September 2026. Whether and which obligations apply to a specific system is a matter for lawyers.

KEY TAKEAWAY

The AI Act regulates purposes, not technologies. Building inventory, record and oversight properly gives the legal assessment a solid foundation.

QUESTIONS FOR YOUR ORGANISATION

6.1 Why can the same agent fall into different risk tiers depending on its use?

6.2 When can a company that merely deploys an AI system end up in the provider role?

6.3 Which dates up to 2028 are still relevant for a company with back-office agents?

EPILOGUE

One year later

Twelve months after that Monday in February, Miriam's work queue looks different.

The straight-through processing rate is 88 per cent. On average, 480 invoices a month land in the queue, and most of them really are difficult. Today Sabine Kraus mainly handles price deviations and clarifications with suppliers – work that requires knowing the suppliers. Two colleagues have moved to cash management, and Aurelia uses early-payment discounts more consistently than ever before.

Not everything went smoothly. In June, for two weeks, the agent noticeably often matched partial deliveries incorrectly after a large supplier had changed its invoice layout. The evals would have shown it sooner had new layouts found their way into the golden set faster. Since then there has been a fixed date for that every month. When the model provider announced a new version in the autumn, it first ran in parallel for three weeks against what were by then 1,900 test cases before it took over the accounts payable agent.

And the robot from October? It still collects invoices from two portals every night. If it stops, the agent is now the first to notice – and reports it to a queue that someone reads.

When a fellow managing director asks Karl Weninger what AI has actually brought, he thinks for a moment. "Less typing, more deciding," he says. "And we know at any time who did what."

What remains

01 **Start with the process, not the technology.** Process mining and metrics show where judgement is needed.

02 **Delegating goals means rearranging responsibility.** An agent is a delegation relationship.

03 **Limits belong in the tool layer.** What is enforced there, holds.

04 **Autonomy is earned level by level.** Backed by evals, a record and the exception rate.

05 **Technology supplies the facts, lawyers the assessment.** A well-kept profile connects the two.

APPENDIX

Summary, notes, further reading

A · The cheat sheet in six sentences

01 An agent is given a goal instead of a sequence of steps and works in a loop until the goal is reached or a rule stops it.

02 Interfaces where possible; RPA where nothing else works; an agent where judgement is needed – but never without real test cases and people who own the exceptions.

03 Model, context, tools and evals are the building blocks; the model alone guarantees nothing.

04 Identity, policy, audit and oversight keep the agent in check.

05 Autonomy rises level by level from L0 to L4, backed by evals, audit trail and exception rate.

06 The EU AI Act classifies by purpose and risk; most back-office agents sit in the lower tiers.

The visual summary is at botforce-technology.com/en/glossary/cheat-sheet.

B · Notes on the questions

CHAPTER 1

1.1 The criterion is who triggers the action in the system. Chatbot: answers the question about a discount period. Copilot: proposes the matching order, a person posts. Agent: matches on its own, posts within its limits and documents the decision.

1.2 Perceive: recognise the input and the gap. Reason: work out where the missing information lies. Plan: define steps and checks. Act: carry them out through tools. Verify: check the result against the criteria, record the reasoning, escalate when unsure.

1.3 Information asymmetry – the principal does not see every step: audit trail. Conflicting goals – the model does not automatically pursue the company's goals: policy and evals. Control: human oversight with escalation and a stop switch.

CHAPTER 2

2.1 RPA needs no interfaces and is therefore quick to introduce. Because it operates user interfaces and replays a fixed script, it breaks on changed screens and unforeseen cases.

2.2 Individual answer. Criteria: interfaces, describability of rules, variants, data structure, exceptions, running costs, typical risk. Often a combination of forms results.

2.3 Process mining shows real flows, exception patterns and how often they occur. From this follows which share of manual cases an agent can address and how far the straight-through processing rate could rise. The patterns also supply test cases for the evals.

CHAPTER 3

3.1 Language models produce plausible answers and fill missing facts with frequent patterns from their training data. RAG places the relevant evidence in the context, making statements justifiable and verifiable.

3.2 An agent acts only through tools. Limits enforced there cannot be overridden by inputs, and all calls can be logged centrally.

3.3 Real historical cases with known correct outcomes, representative and including the important exception patterns, versioned and continuously extended with new error cases.

CHAPTER 4

4.1 A prompt is an instruction to a probabilistic model and can be misunderstood or overridden by inputs. Only a limit enforced in the tool layer works reliably and can be demonstrated to auditors.

4.2 Identity: the agent acts under its own, tightly scoped identity. Policy: the €10,000 limit prevents the autonomous posting. Oversight: the case lands in Sabine's queue with proposal and reasoning. Audit: four log lines document the process with timestamps.

4.3 Transfer directly: segregation of duties, four-eyes principle, complete records, access reviews. Adaptation: because of pace and volume, controls must work completely and automatically rather than as spot checks.

CHAPTER 5

5.1 At L3 a person confirms every action; at L4 the agent acts within limits and people decide only exceptions. The move requires a high agreement rate, stable evals without regression, a complete audit trail and an exception rate the team can handle.

5.2 The level applies per group of cases. Unambiguous, low-risk cases can move to L4 earlier; uncertain or riskier cases stay at L3 until the evidence is sufficient.

5.3 An owner in the business team, planned capacity for the queue, technical operations, people responsible for evals and error review. If a role is missing, exceptions pile up, errors go unnoticed or responsibility remains unclear.

CHAPTER 6

6.1 Classification depends on the intended purpose and on how strongly decisions affect people. Matching invoices is different from pre-screening job applications, which falls under Annex III.

6.2 When it substantially modifies the system or uses it under its own name. The division of roles should be settled in the contract.

6.3 On 2 December 2026 the deadline for machine-readable marking of existing systems expires; high-risk obligations follow on 2 December 2027 (Annex III) and 2 August 2028 (Annex I), if a use falls under them. Transparency obligations and AI literacy already apply. The classification of specific systems must be checked legally.

C · Further reading

Automation & RPA	botforce-technology.com/en/glossary/automation
AI & machine learning	botforce-technology.com/en/glossary/ai
Agentic AI	botforce-technology.com/en/glossary/agentic-ai
EU AI Act	botforce-technology.com/en/glossary/eu-ai-act
Agentic AI cheat sheet	botforce-technology.com/en/glossary/cheat-sheet

SUGGESTED CITATION BOTFORCE GmbH (2026): From script to goal. Agentic AI in the enterprise – a practical guide with a case story. Vienna.

PUBLISHER BOTFORCE GmbH, Vienna · As of September 2026. The case story is entirely fictional. Statements about the EU AI Act are not legal advice.