Welcome to

Knowledge Engineering

Prof. Dr. Knut Hinkelmann
Dr. Emanuele Laurenzi

Introduction to Knowledge-Based Systems

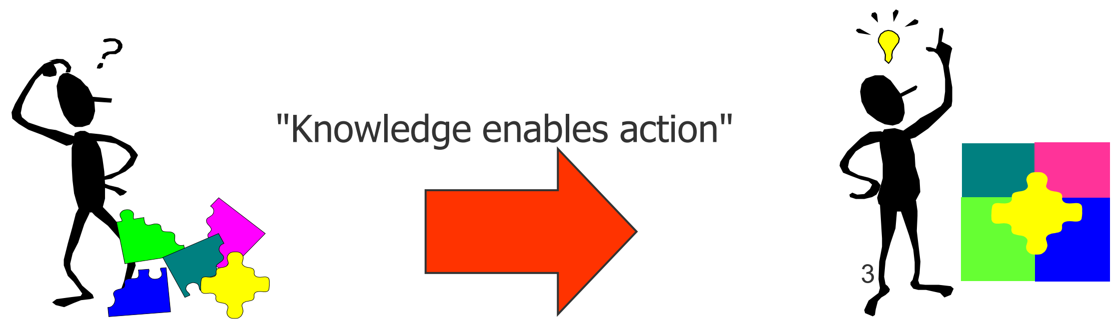
Was ist Wissen?

Philosophy:

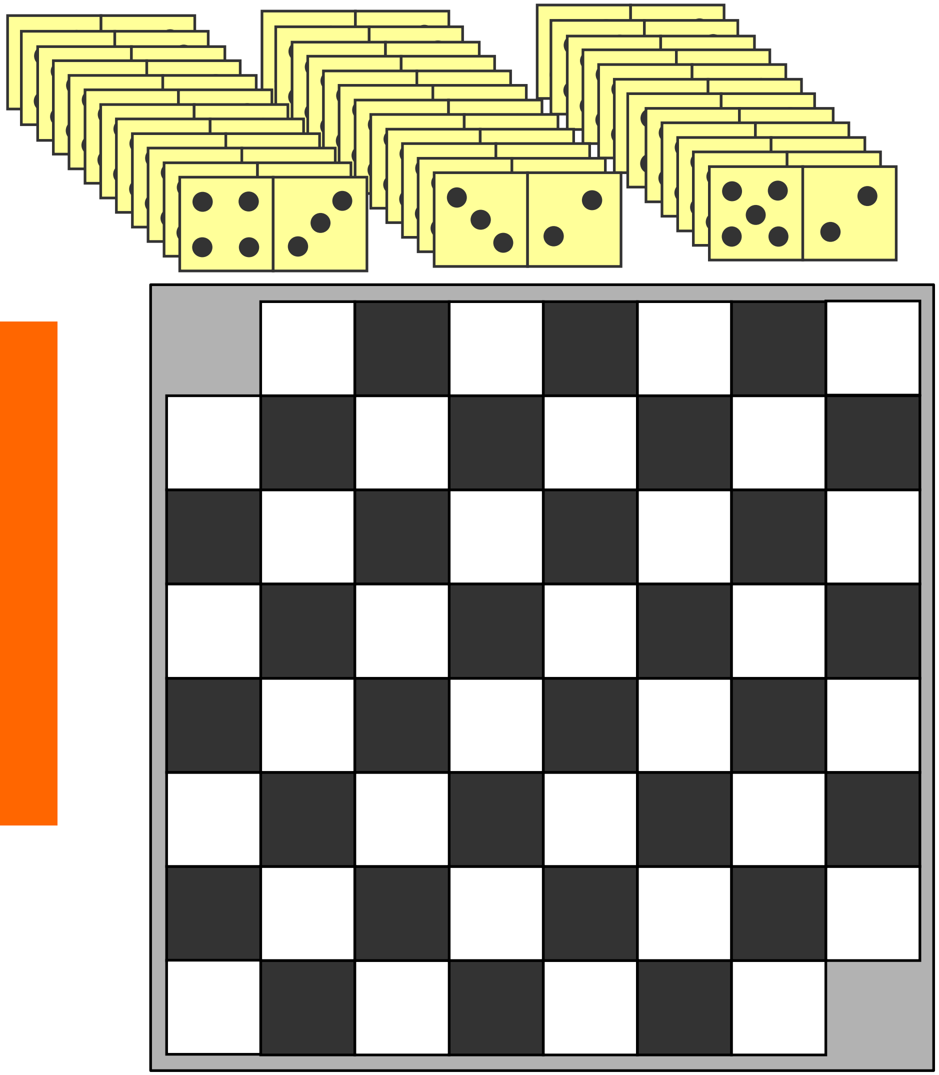
In contrast to opinion and belief, knowledge is based on reasoning and subject to strict verification postulates (justified true belief).

Everyday Kowledge

Generally available orientations in the context of everyday action and factual contexts. Knowledge is closely linked to rationality.



Problem Solving: Example



Placing dominos on a chess board

- Given a chess board where two opposite corners are missing
- A domino covers two adjacent field
- Is it possible to cover all fields of the board with dominos?

Why is knowledge important for companies?

What is the problem with knowledge?

Discussion: What is knowledge work?

- Give examples of knowledge work
- Explain, why you regard this work as knowledge work.

Some categories of knowledge work

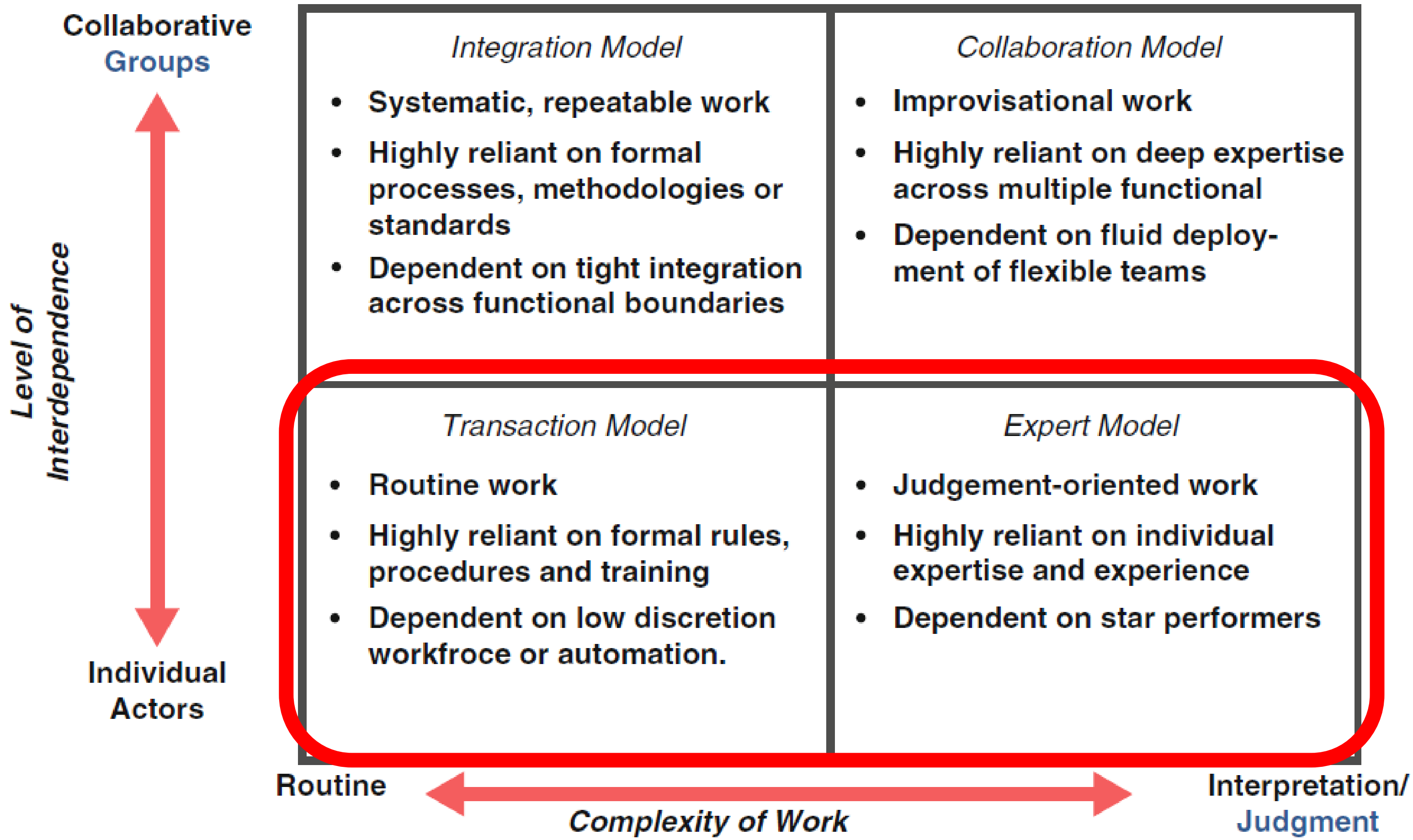
- Decision-Making
  - Making a choice between different alternatives.
- Diagnosis
  - identification of the nature and cause of something, e.g. a disease or a failure in a machine; (can be a prerequisite for solving a problem)
- Problem Solving
  - Finding solutions to a problem satisfying specified goals, e.g. treatment of a disease
- Configuration
  - special case of design activity, where the artifact is assembled from instances of a fixed set of component types
- Planning
  - organizing activities to achieve a desired goal
- Design
  - construction of an artifact (object or a system), satisfying a set of requirements, subject to constraints

Application of Knowledge

Examples from a Car Rental Company

- Decision-Making
  - Choose between different offers for new cars
- Diagnosis/Problem Solving
  - Find the failure if the engine of the car does not start
- Configuration
  - Select equipment for new cars
- Planning
  - Scheduling of cars so that they are at the branch
- Information Retrieval
  - Find all documents with regulations about international drivers licences

Types of Knowledge Work according to (Davenport 2010)



(Davenport 2010)

Process-orientation for Knoweldge Workers according to (Davenport 2010)

Transaction workers. Need to understand the flow of their work and the knowledge needed to perform it, but rarely have time to consult guidelines or knowledge sources. Process flow can be added into IT applications (workflows) bringing required information to the worker.

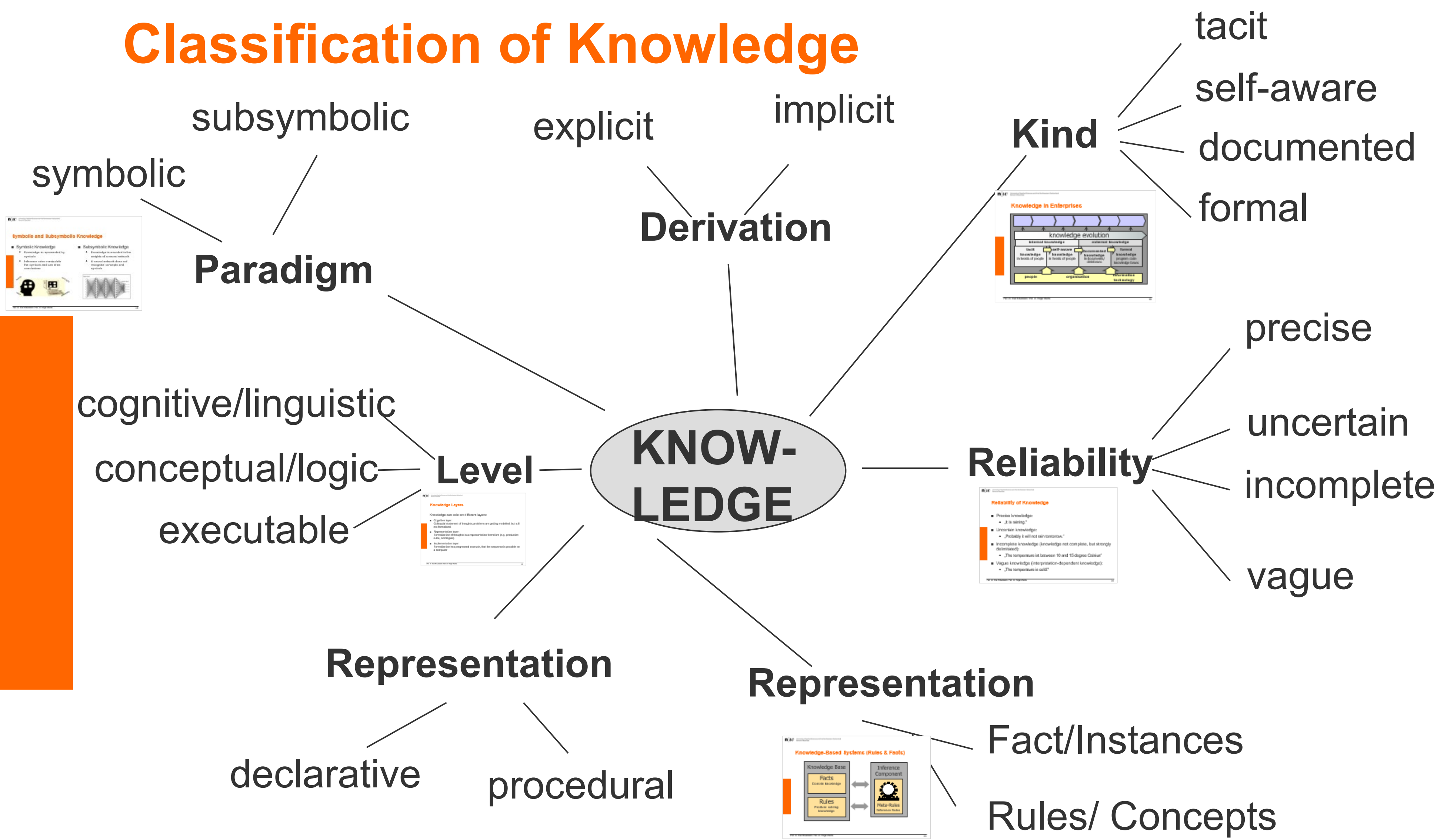
Integration workers. It is possible to articulate the process to be followed, e.g. by "standard operating procedures". Workers typically have enough time and discretion to consult the description.

Expert workers. High autonomy and discretion in the work. Expert knowledge work can be improved by providing templates, sample outputs, and high-level guidelines instead of specifying detailed process models.

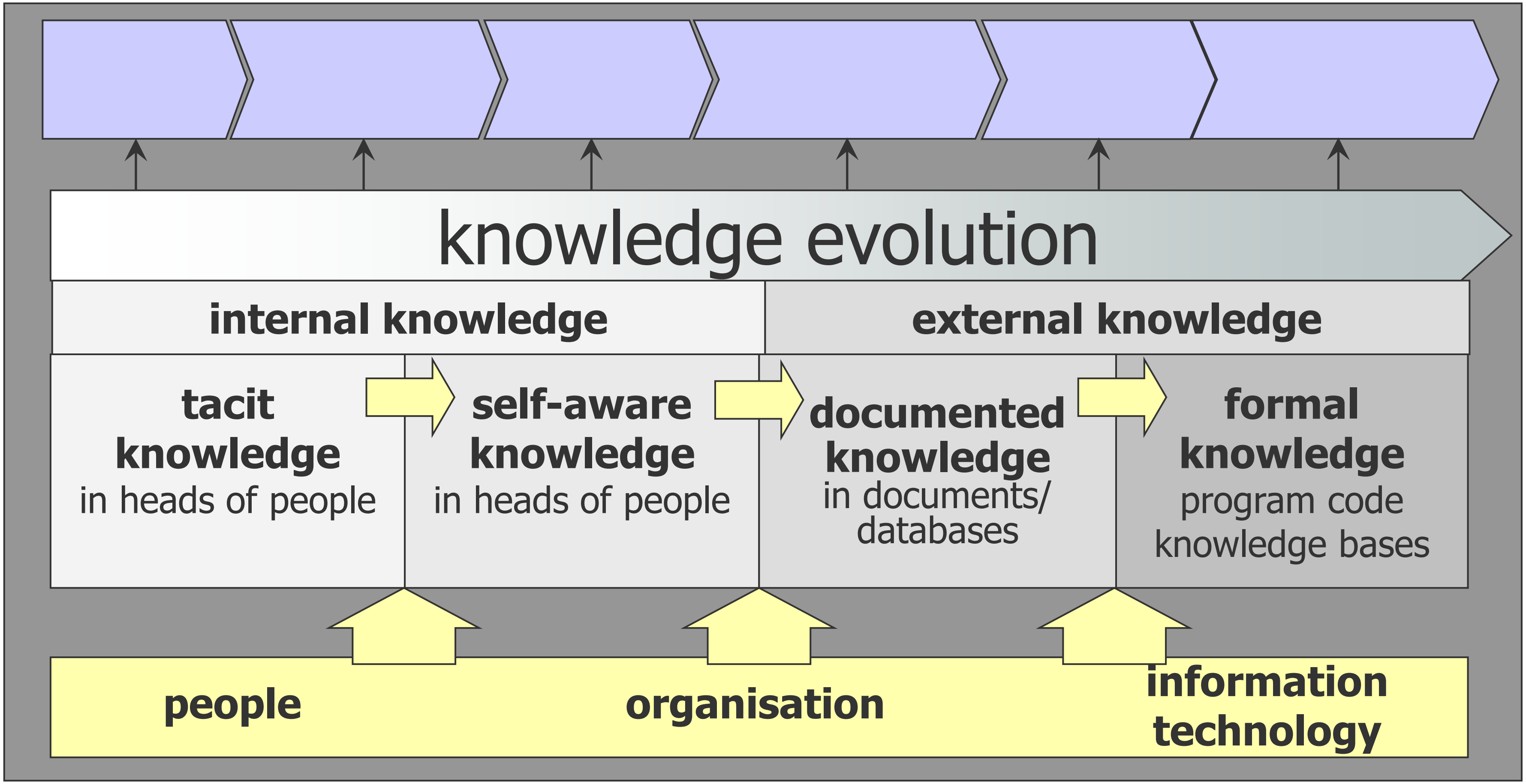
Collaboration workers. If external knowledge and information are necessary to do the job, they must generally be made available through repositories and documents

(Davenport 2010)

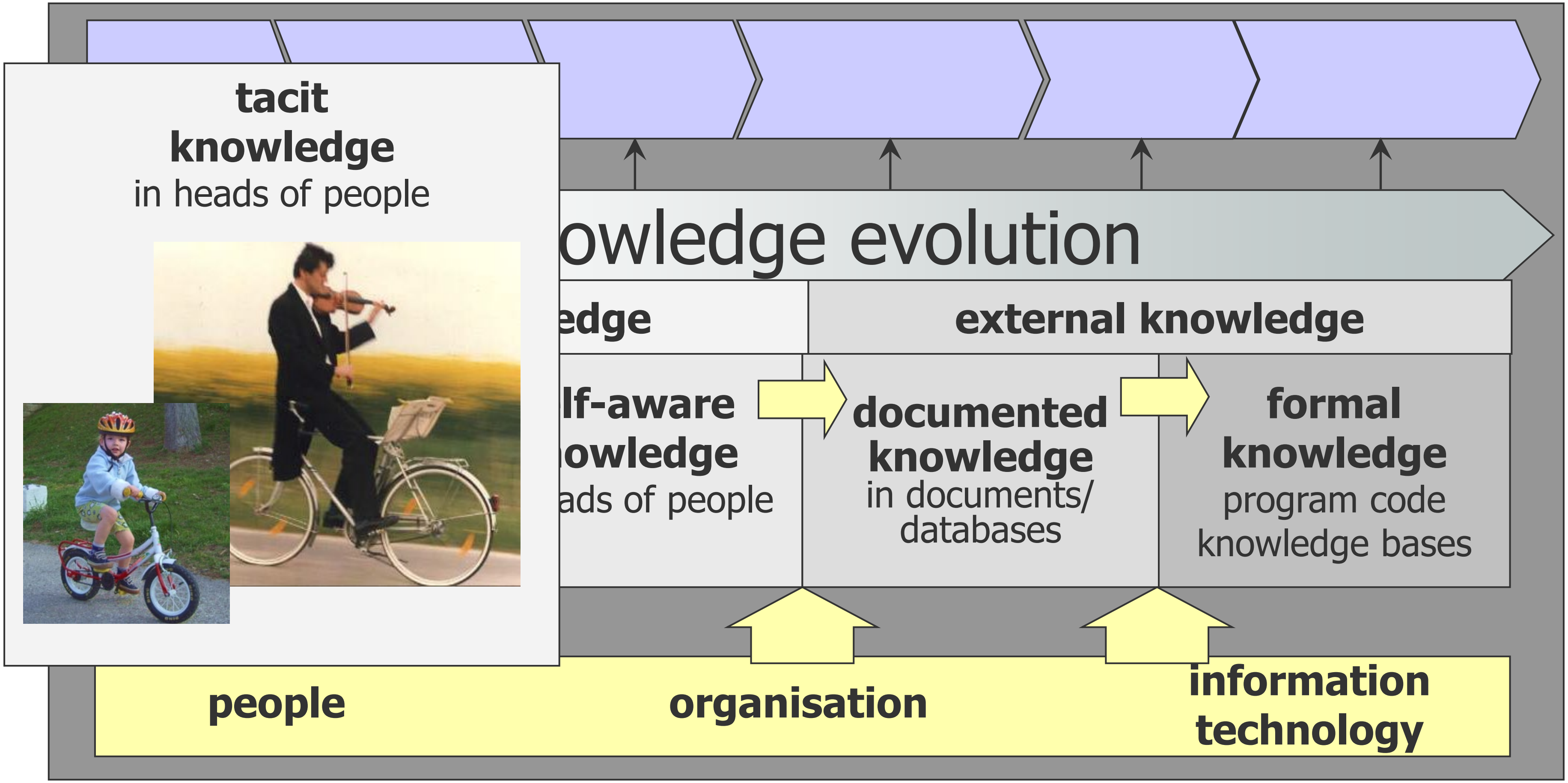
Classification of Knowledge



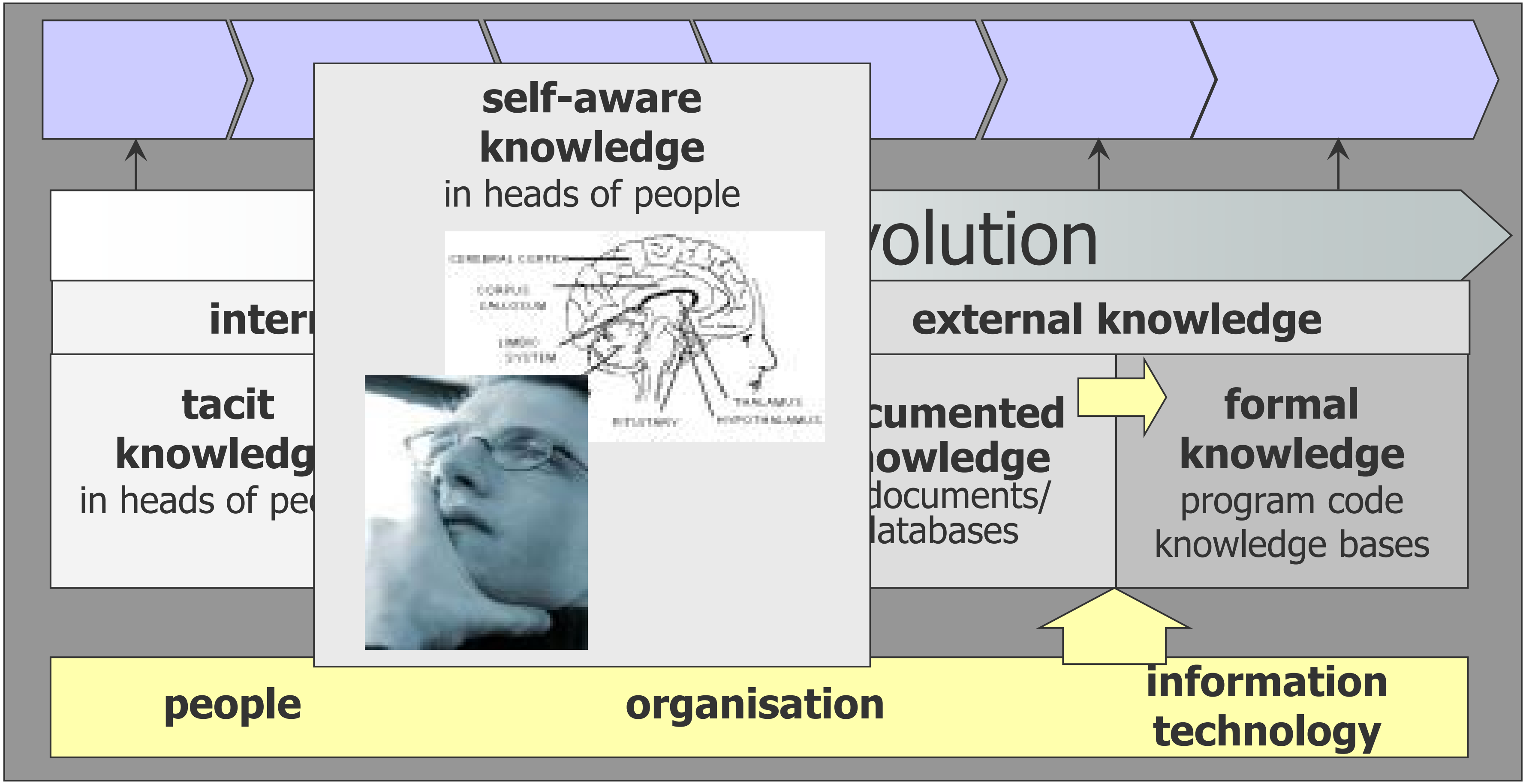
Knowledge in Enterprises



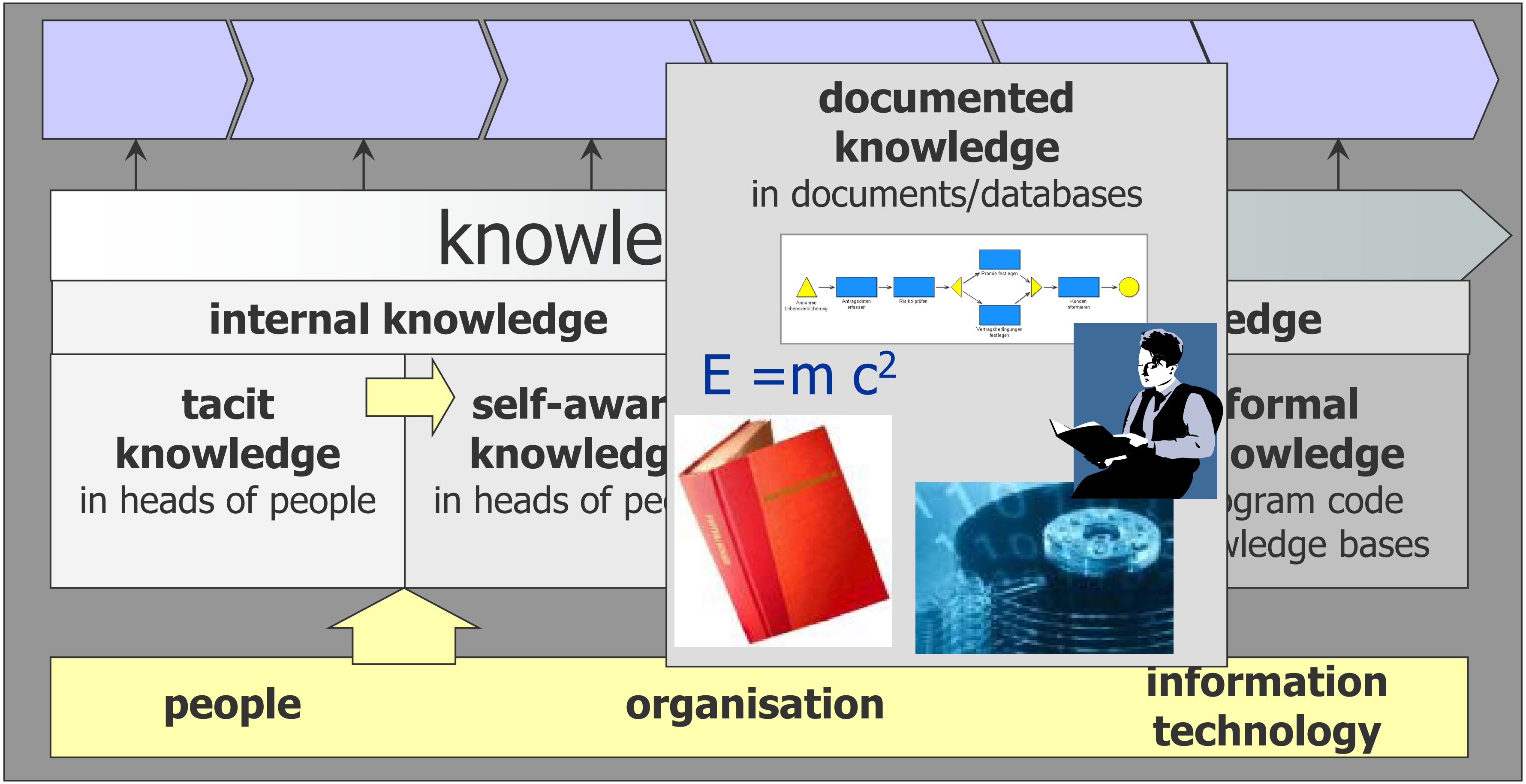
Knowledge in Enterprises



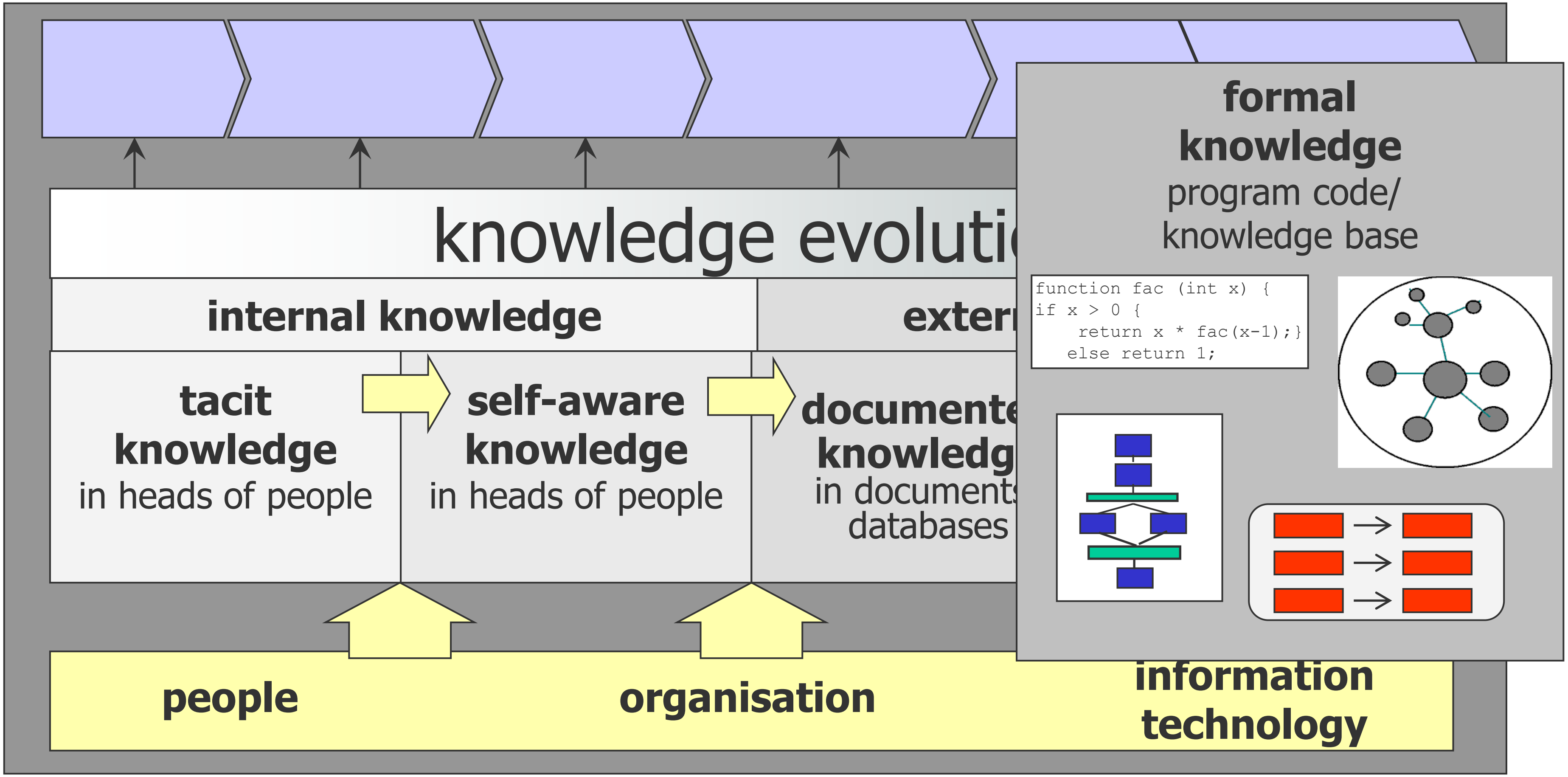
Knowledge in Enterprises



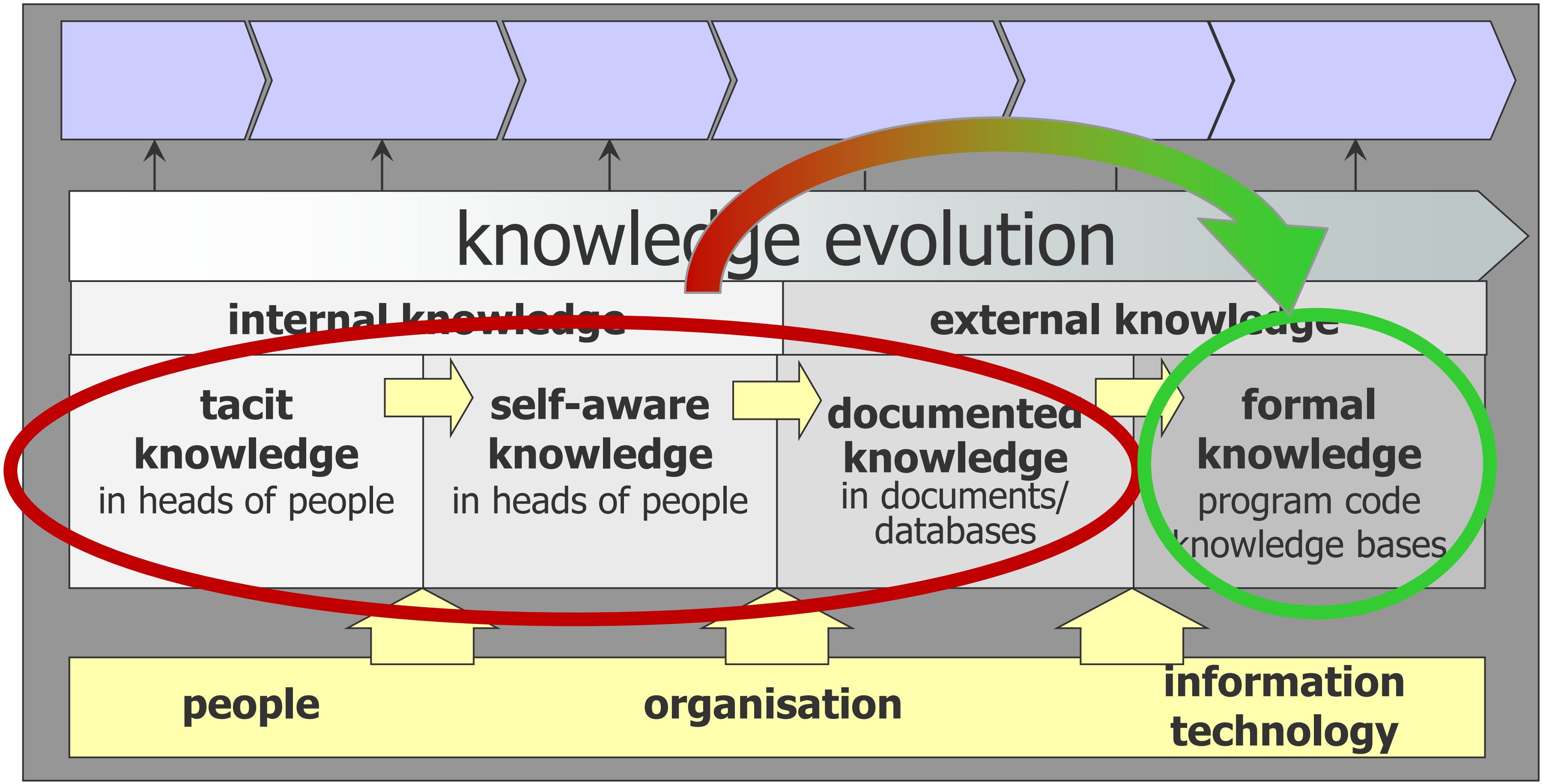
Knowledge in Enterprises



Knowledge in Enterprises



Objective: Formalize Knowledge for Digitalization

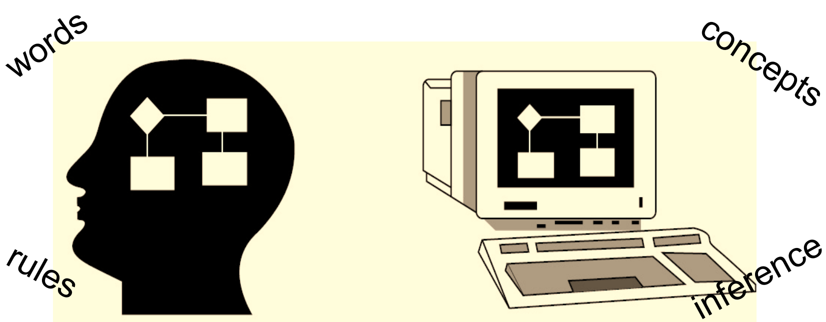


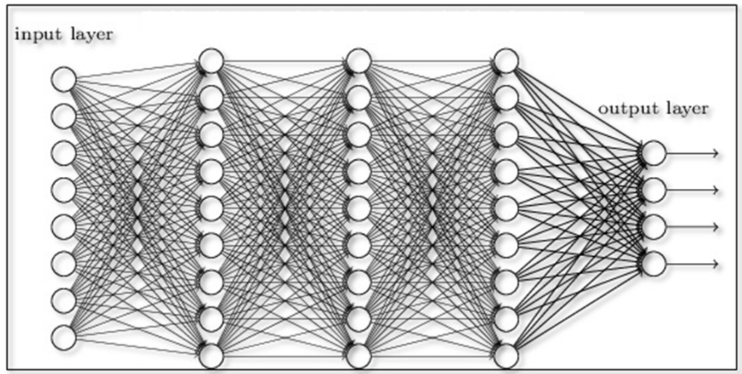
Reliability of Knowledge

- Precise knowledge:
  - „It is raining."
- Uncertain knowledge:
  - „Probably it will not rain tomorrow.“
- Incomplete knowledge (knowledge not complete, but strongly delimitated):
  - „The temperature ist between 10 and 15 degree Celsius“
- Vague knowledge (interpretation-dependent knowledge):
  - „The temperature is cold."

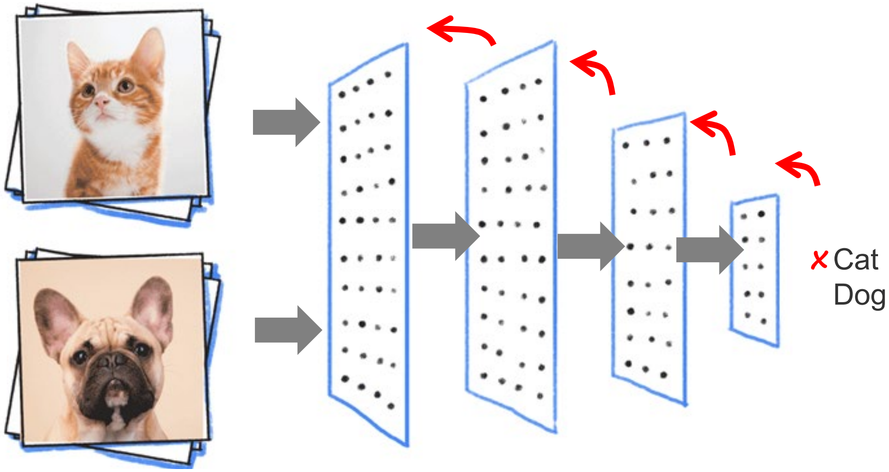
Symbolic and Subsymbolic Knowledge

- Symbolic Knowledge
  - Knowledge is represented by symbols
  - Inference rules manipulate the symbols and can draw conclusions
- Subsymbolic Knowledge
  - Knowledge is encoded in the weights of a neural network
  - A neural network does not recognise concepts and symbols





Subsymbolic Knowledge: Neural Network

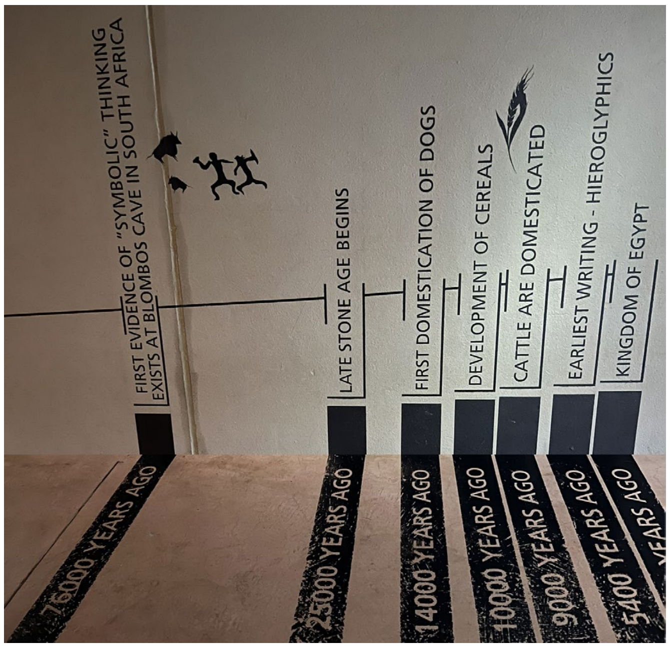


Application: Cat or Dog?

- Knowledge Processing
  - Representation: units, weights between units
  - Derivation of knowledge: Connotation

Physical Symbol System Hypothesis

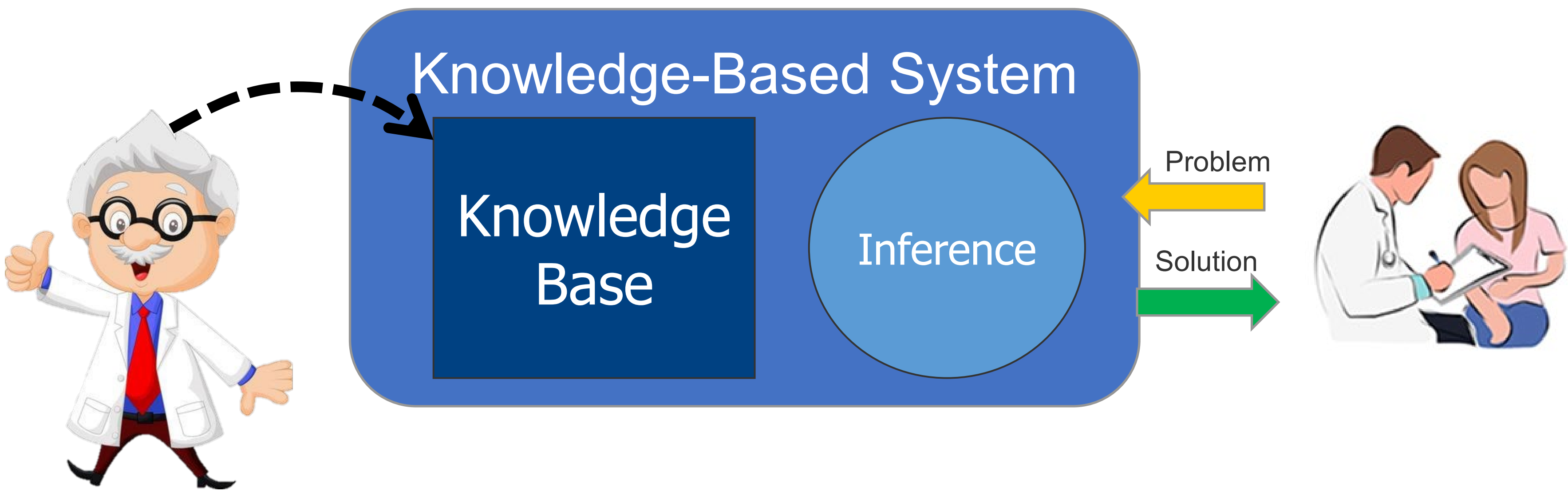
The ability to think symbolically
is a characteristic of intelligence



> "A physical symbol system has the necessary and sufficient means for general intelligent action"
>
> Allen Newell and Herbert A. Simon

Knowledge-Based System

- *„An Expert System is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require human expertise for their solutions. “* (Feigenbaum 1982)
- The term „knowledge-based systems“ is often used synonym for „expert systems“. It makes clear that the system has an explicit symbolic knowledge base.



Paradigms of Symbolic Knowledge Processing

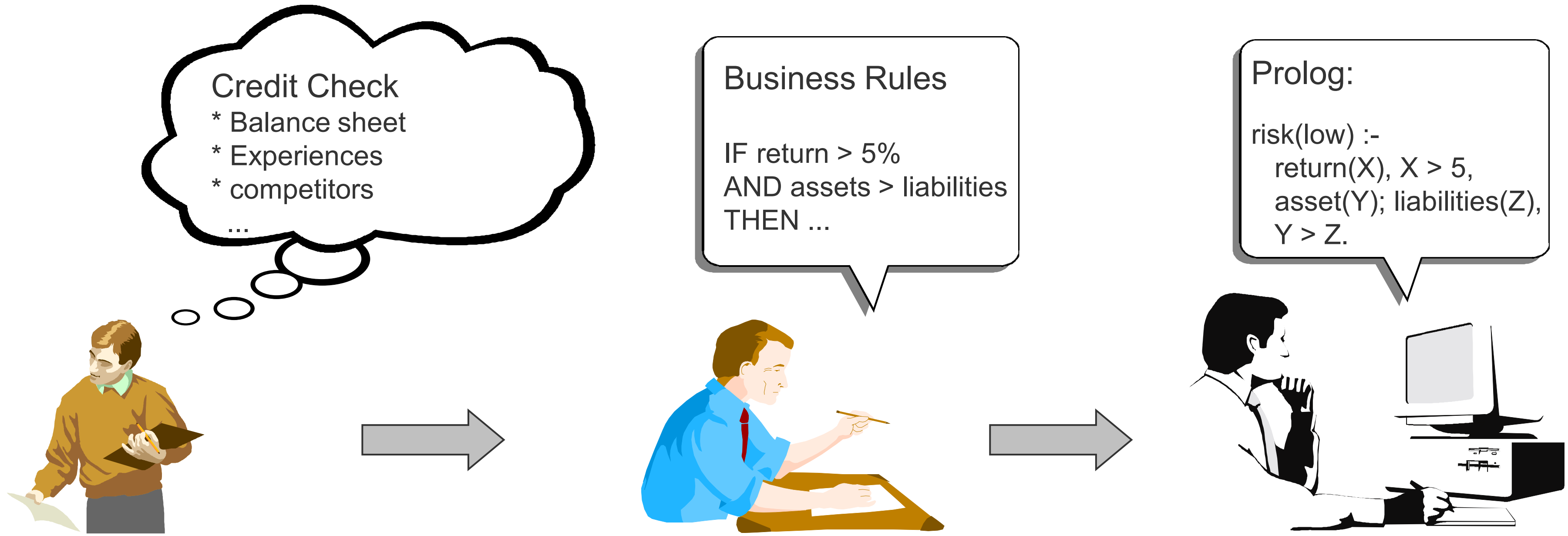
- Logic Systems:
  - Representations: logical formulas
  - Processing: Logical inference, deduction
- Non-Logic Systems:
  - Representations: condition-action rules
  - Processing: Action, modification of knowledge base
- Fuzzy Systems:
  - Representation: linguistic formulated knowledge
  - Processing: Approximate conclusion

Knowledge Layers

Knowledge can exist on different layers:

- Cognitive layer:
  Colloquial statement of thoughts; problems are getting modelled, but still not formalised.
- Representation layer:
  Formalisation of thoughts in a representation formalism (e.g. production rules, ontologies)
- Implementation layer:
  Formalisation has progressed so much, that the sequence is possible on a computer

Layers of Knowledge-Based Systems

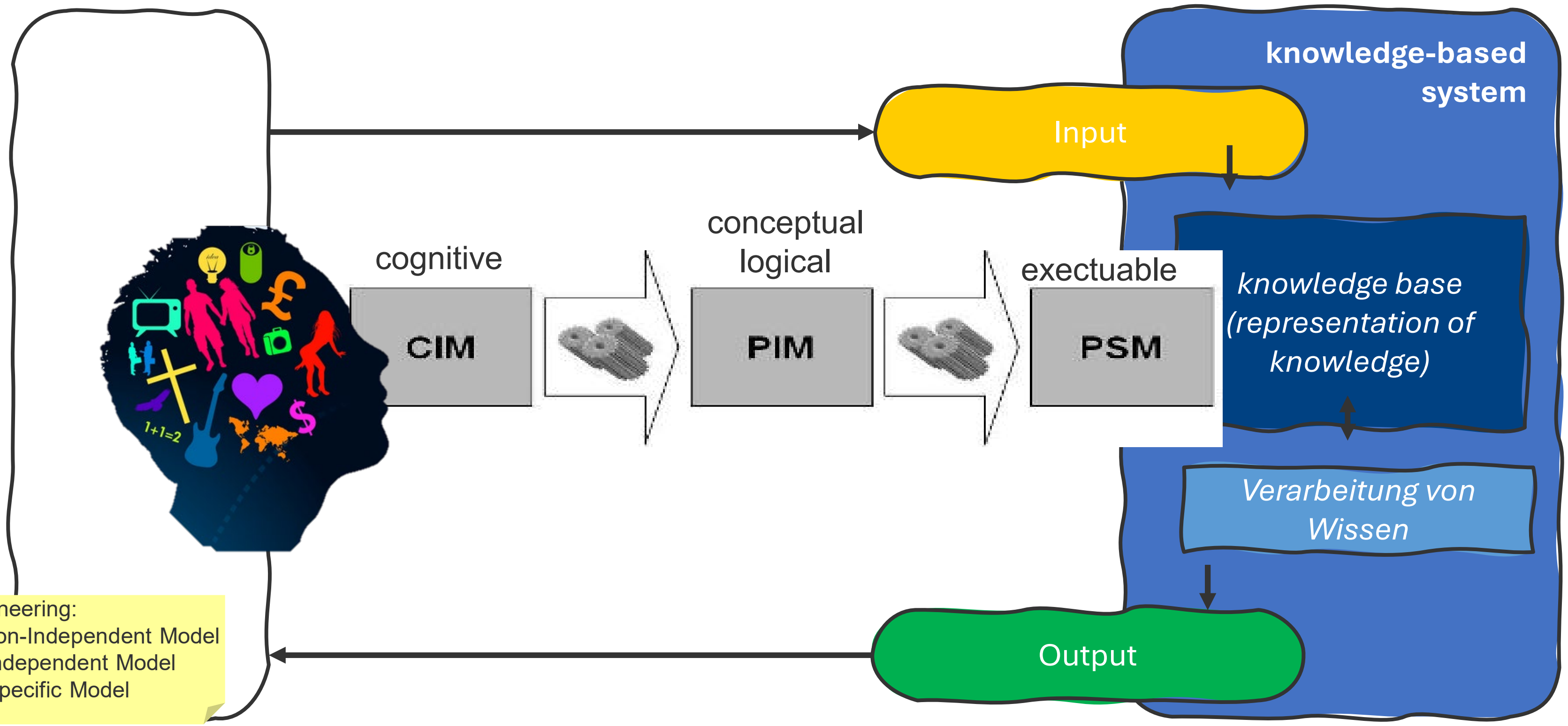


Cognitive/linguistic ↓ Any concepts, terms, expressions

Conceptual/logical ↓ Definition of concepts, relations, rules.

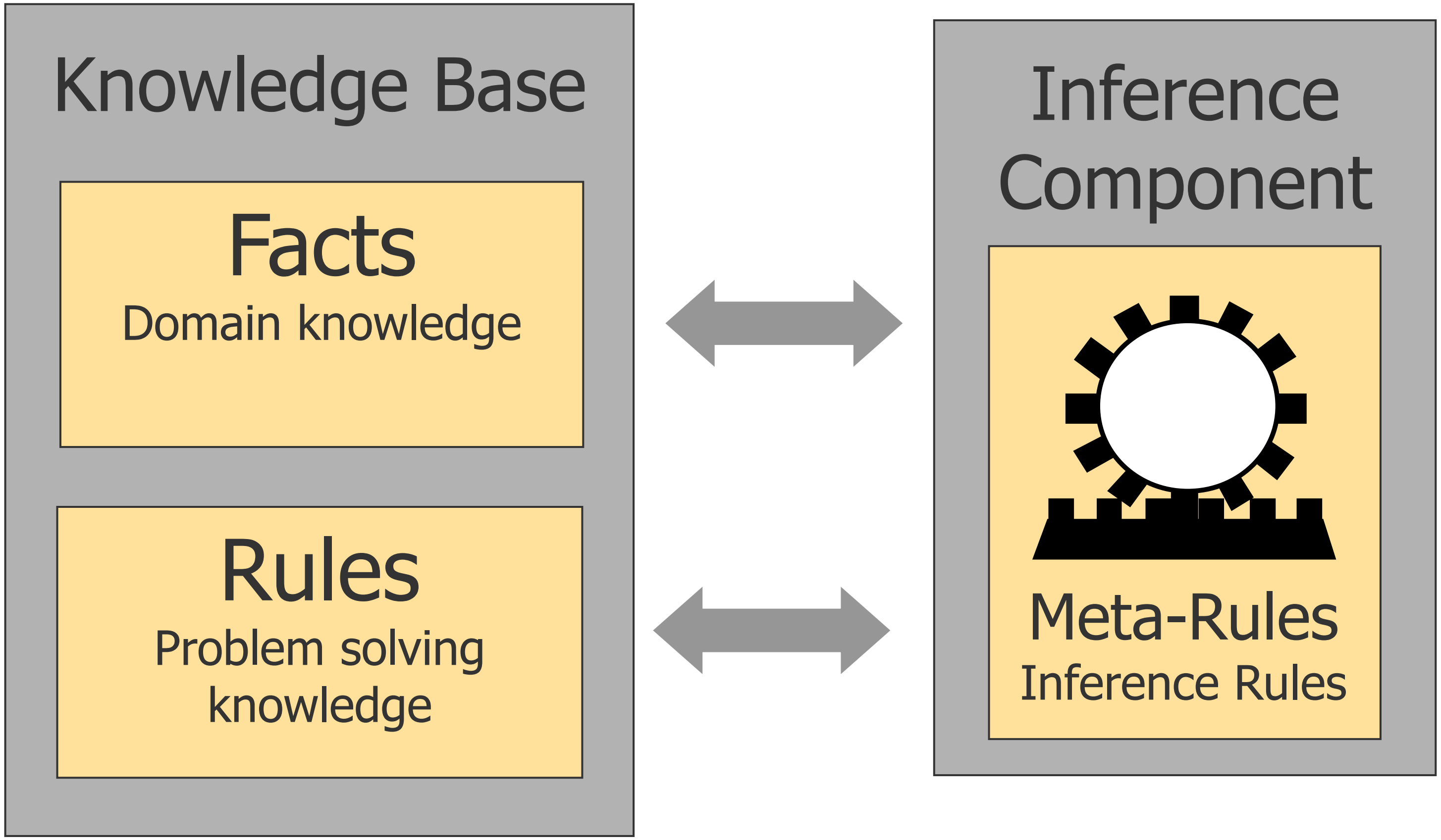
executable ↓ Represented in executable formalism

Knowledge Engineering: Building a Knowledge Base



Model-driven Engineering:
CIM = Computation-Independent Model
PIM = Platform-Independent Model
PSM = Platform-Specific Model

Knowledge-Based Systems (Rules & Facts)



Types of Knowledge

- **Facts**: statements about reality
  - Example: *Socrates is human*
- **Rules**: General proposition about relations or procedure that are valid under specific conditions (e.g. in an „if ... then“-form“)
  - Example: *All humans are mortal*

Derivation

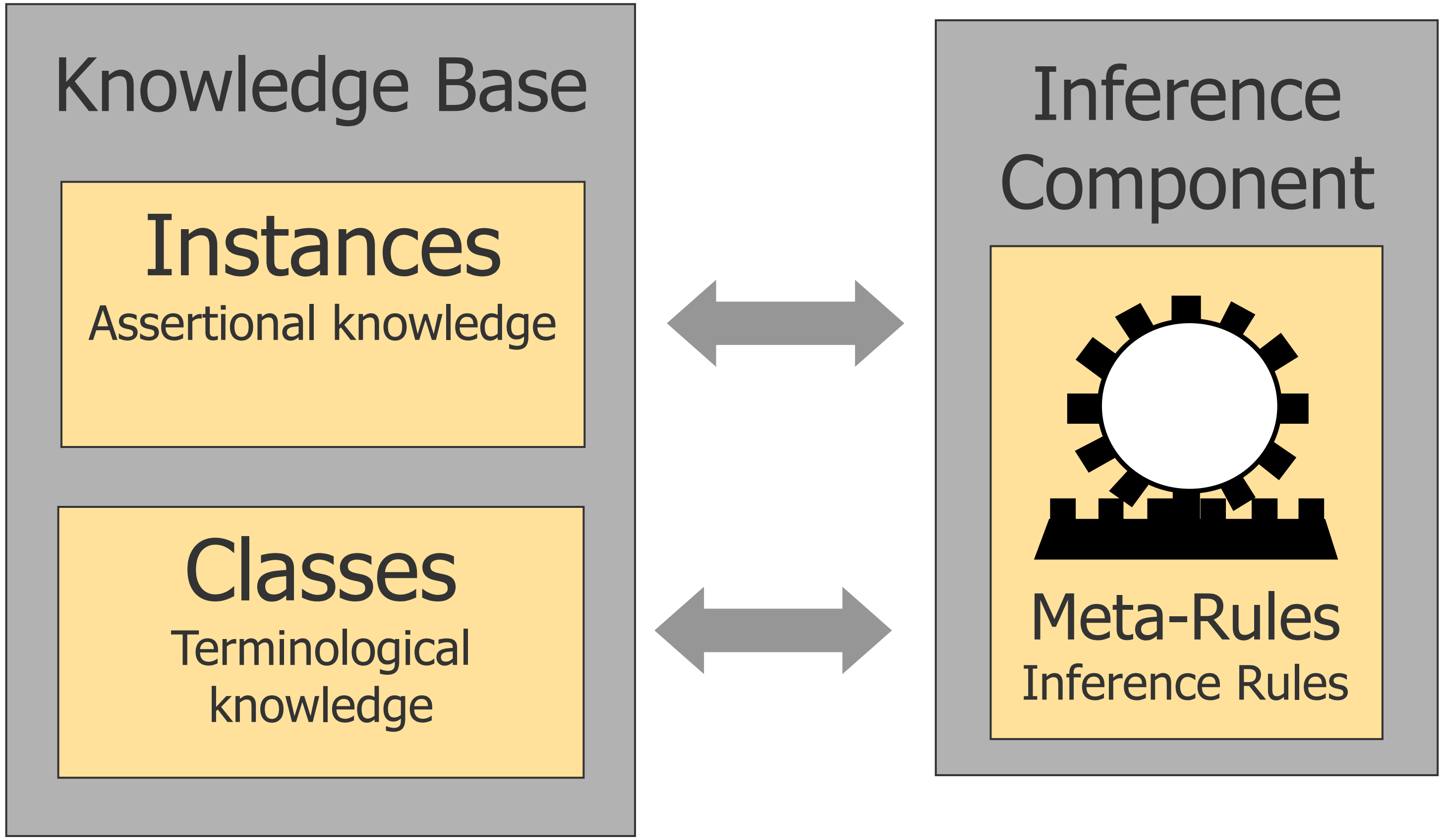
- Explicit knowledge:
  - knowledge which is stored in the knowledge base (static knowledge)
- Implicit knowledge:
  - not explicitly stated in the knowledge base
  - is determined from facts by application of rules
- *Derivation = Inference = Reasoning*
  - New knowledge is generated from existing one: Making implicit knowledge explicit

Socates is human.
All humans are mortal.

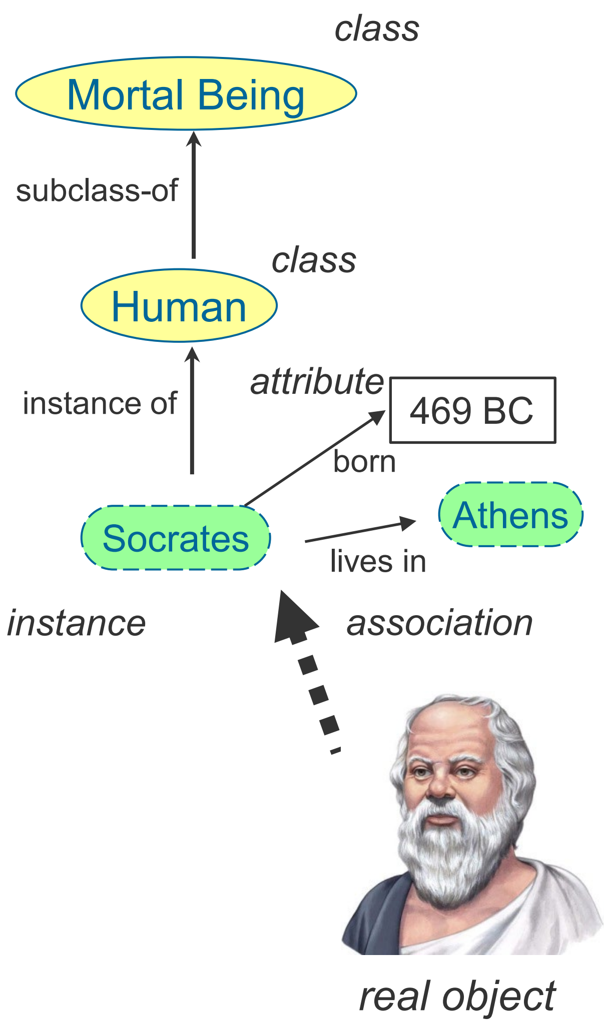
→ deductive inference →

Socrates is mortal.

Knowledge-Based Systems (Classes and Instances)



Concepts, Instances and Relations



- There are two kinds of concepts:
  - **generic: classes**
  - **individual: instances**
- There are different kinds of relations
  - **Structural**
    - between classes **(subclass of)**
    - between instance and class **(instance of)**
  - **Non-structural:** any other kind of relationship
    - **Object Property** (relation): range is a class, i.e. value is an instance of a class
    - **Data Property** (attribute): range is a data type (number, string)

Types of Knowledge

- **Classes:** Groups with common characteristics
  - Example: human, mortal being
- **Instances:** statements about real objects
  - Example: Socrates

Relations:

- Instance of:
  - Socrates is human
    "Socrates is an *element* of the set of all humans"
- Subclass of ("is a"):
  - Humans are specializations (is-a) of Mortal Beings
    "Humans are a *subset* of the set of all mortal beings"

Derivation

- Explicit knowledge:
  - knowledge which is filled away in the knowledge base (static knowledge)
- Implicit knowledge:
  - not explicitly stated in the knowledge base
  - is determined from facts by application of rules
- *Derivation = Inference = Reasoning*
  - New knowledge is generated from existing one: Making implicit knowledge explicit

Socrates is a Human.
Humans are specializations (is-a) of Mortal Beings.

→ deductive inference →

Socrates is a Mortal being.

Meta Rules

- Meta Rules …
  - implement the Inference
  - control the application of rules/concepts
  - are part of the Inference Engine
- Meta Rules can be general, e.g.
  - If all conditions of a rule are satisfied then add the conclusion to the knowledge base
  - If more than one rule can be applied use the most specific one
  - If more than one rule can be applied, use the first one

… or domain specific

  - For underwriting in health insurance, first apply the rules that deal with the health conditions and then check for the credibility of the applicant

Example of a Declarative Knowledge Base

father(peter,mary)
father(peter,john)
mother(mary,mark)
mother(jane,mary)

---

father(X,Y) AND father(Y,Z) → grandfather(X,Z)
father(X,Y) AND mother(Y,Z) → grandfather(X,Z)
mother(X,Y) AND father(Y,Z) → grandmother(X,Z)
mother(X,Y) AND mother(Y,Z) → grandmother(X,Z)
father(X,Y) AND father(X,Z) AND Y =/= Z → sibling(Y,Z)
mother(X,Y) AND mother(X,Z) → sibling(Y,Z)

The rules can be used to

- Derive all grandparent and sibling relationships (forward chaining)
- Answer questions about relationships (backward chaining)

Example of a Declarative Knowledge Base

father(peter,mary)
father(peter,john)
mother(mary,mark)
mother(jane,mary)

---

father(X,Y) → parent(X,Y)

mother(X,Y) → parent(X,Y)

father(X,Y) AND parent(Y,Z) → grandfather(X,Z)

mother(X,Y) AND parent(Y,Z) → grandmother(X,Z)

parent(X,Y) AND parent(Y,Z) → grandparent(X,Z)

parent(X,Y) AND parent(X,Z) → sibling(Y,Z)

Declarative vs. Procedural Knowledge

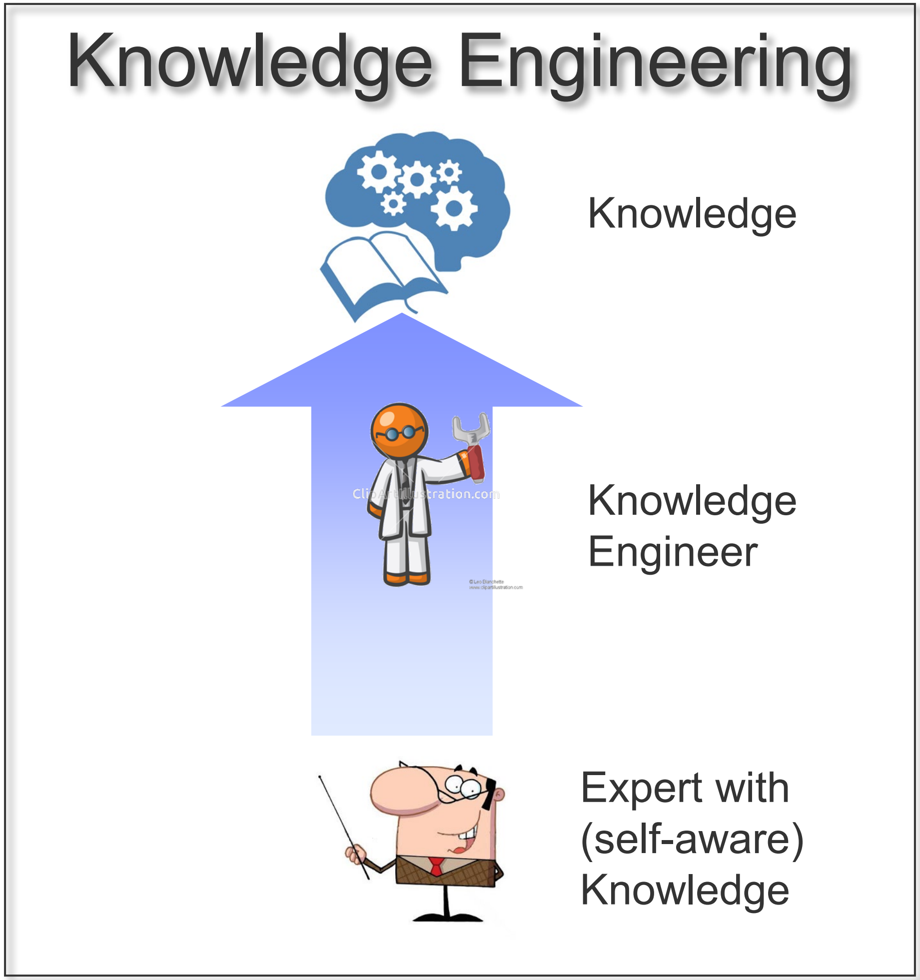
- *Declarative knowledge*: The representation of knowledge is independent of an inference engine
- *Procedural knowledge*: The representation of knowledge determines its use, e.g. representing actions, order/flow of tasks, updating knowledge

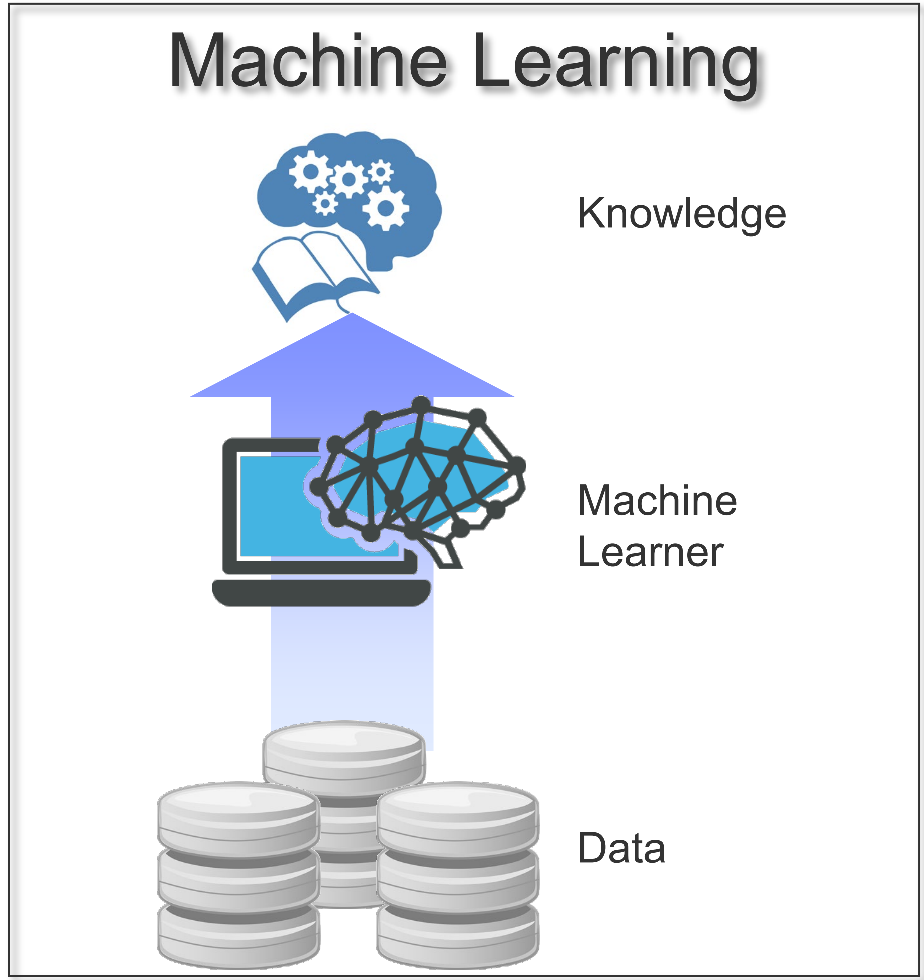
  if a car reaches the traffic light
  and the traffic light has switched to red
  then hold at the stop line

  if account balance is X
  and deposit is Y
  then account balance is X + Y

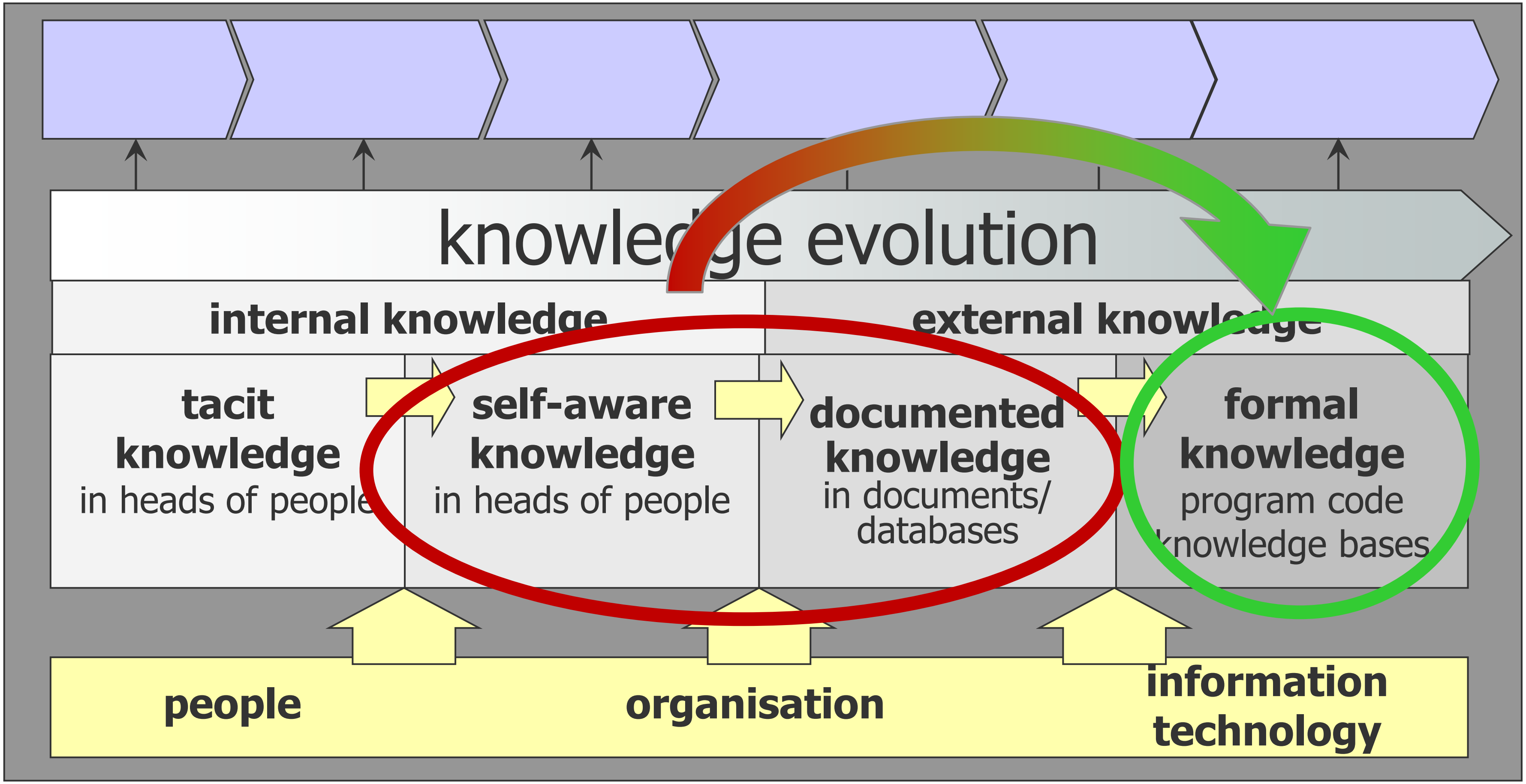
How can you acquire the knowledge?

Acquiring Knowlegde

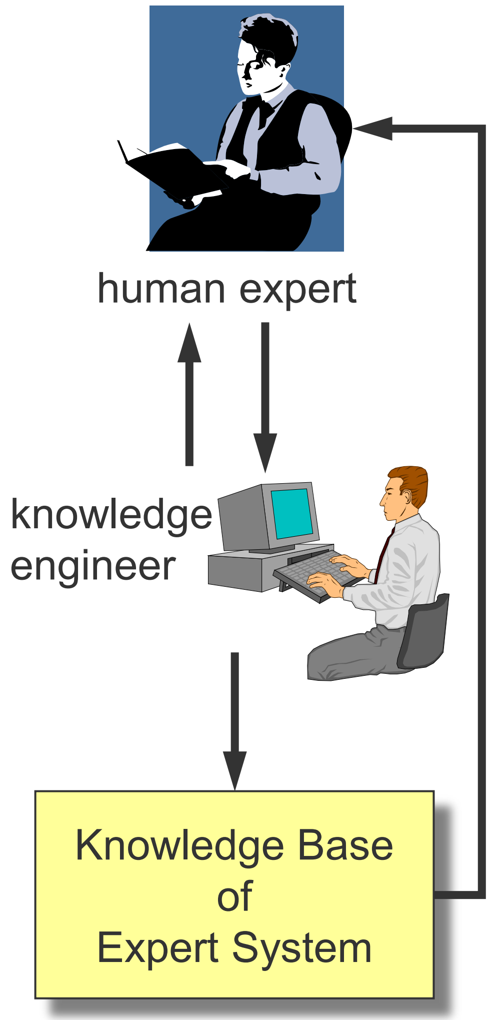




Knowledge Engineering



Knowledge Engineering

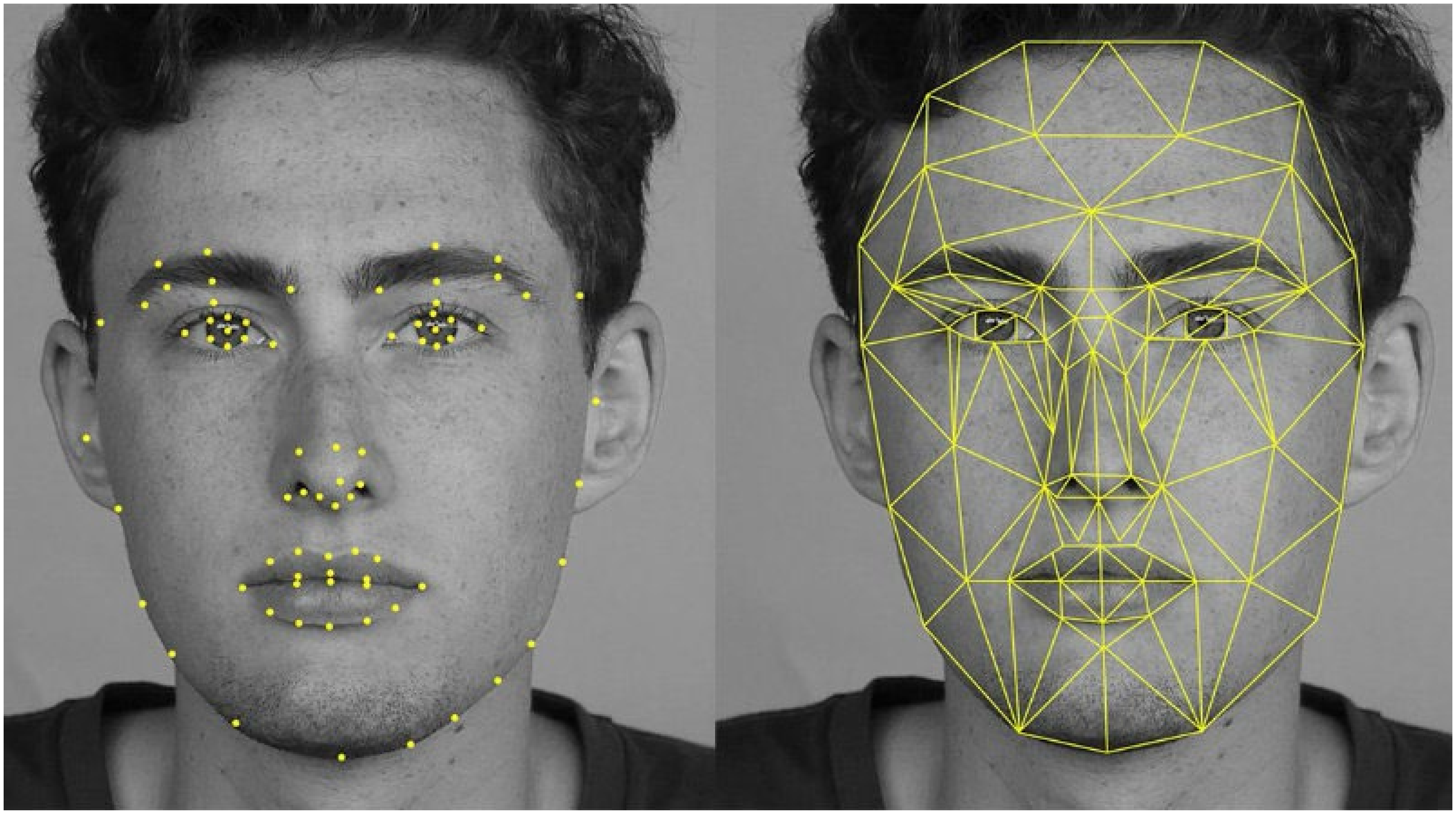


- Knowledge Engineering is the process of
  - building and
  - maintaining

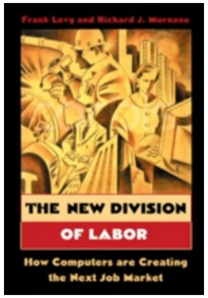
  knowledge-based systems or intelligent agents
- *"Knowledge Engineering is an engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise."*[1)]
- Sources of knowledge
  - Human experts
  - Documentation

1) Feigenbaum, E., and P. McCorduck. (1983). The Fifth Generation. Reading, MA: Addison-Wesley

Face Recognition



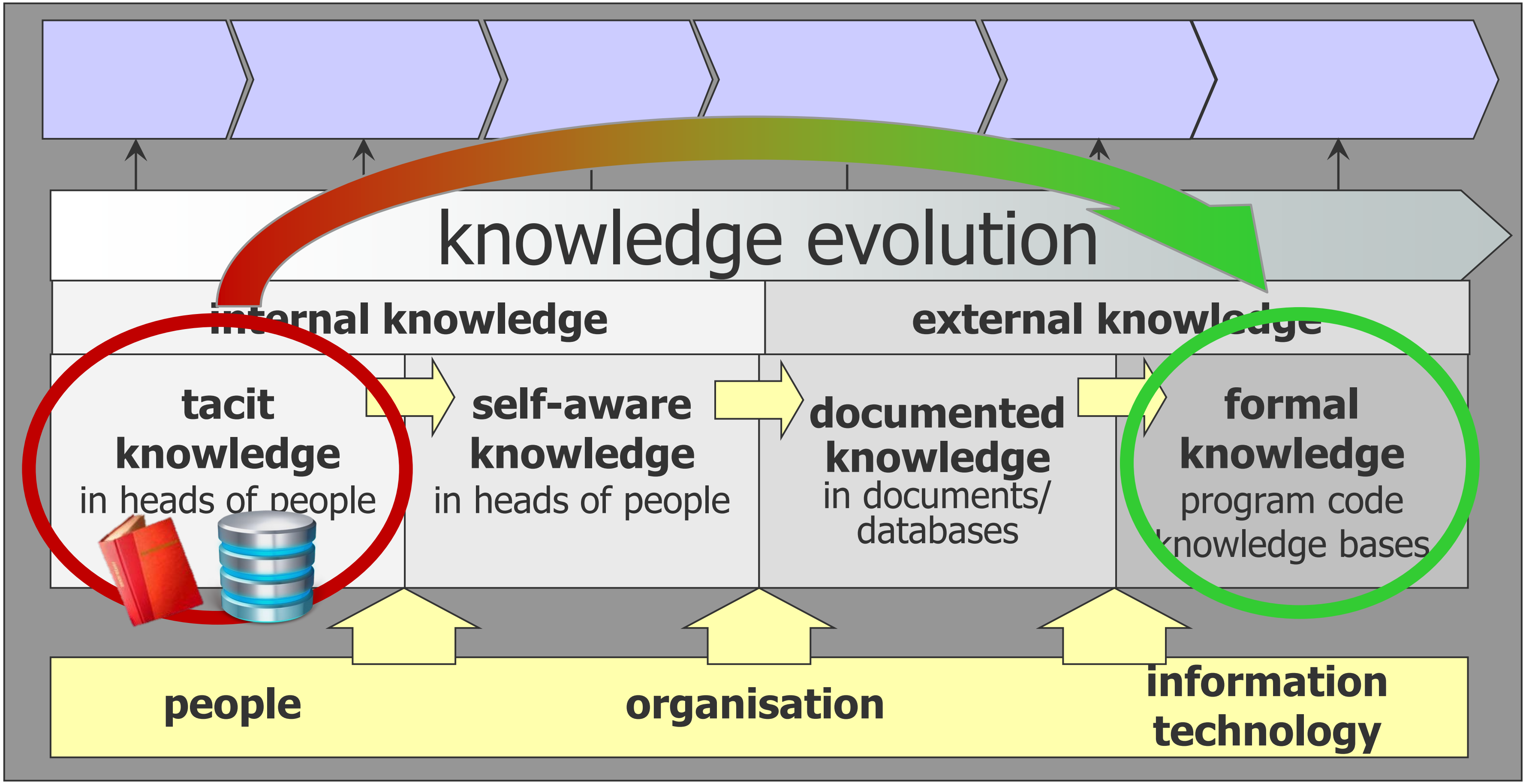
Self-driving Cars



"… it is hard to imagine discovering the set of rules that can replicate the driver's behavior."

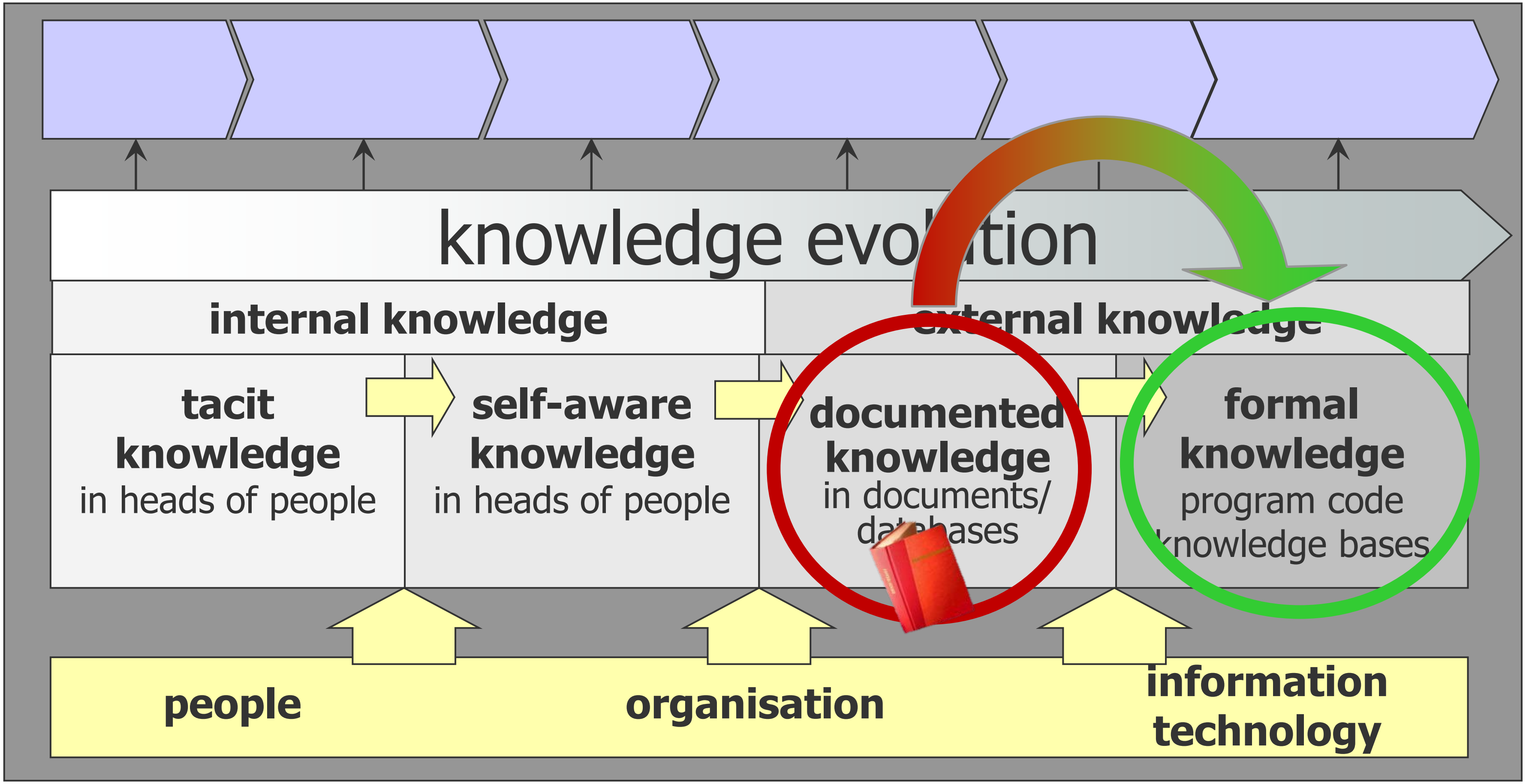
(Levy & Murnane 2006)

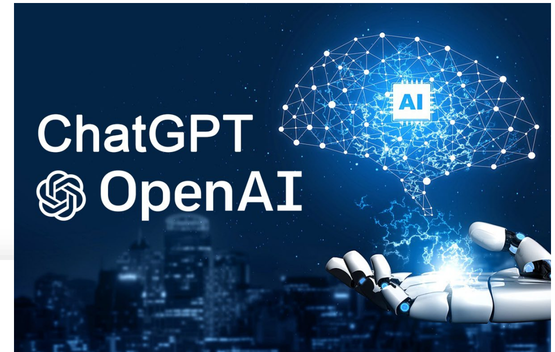
Machine Learning: Make Knowledge explicit with the Use of Data



From data (texts or structured data) it is possible to learn tacit knowledge and new knowledge

Machine Learning:
Learning from Documents





chatGPT learned from Documents

Maybe I should ask chatGPT why I should study knowledge engineering

KN Why should I study knowledge engineering?

There are several reasons why you might consider studying knowledge engineering:

1. Demand: Knowledge engineering is a growing field that is in high demand, particularly in industries such as artificial intelligence, data analytics, and machine learning. As companies increasingly rely on data to inform decision-making, there is a need for professionals who can design and implement knowledge-based systems.
2. Integration of Knowledge: Knowledge engineering involves the integration of knowledge from multiple sources, including human experts and machine learning algorithms. This process is critical for developing accurate and reliable knowledge-based systems that can be used to solve complex problems.
3. Interdisciplinary Field: Knowledge engineering is an interdisciplinary field that draws on knowledge from computer science, cognitive science, and philosophy. As such, it provides an opportunity for students to develop a broad range of skills and knowledge that can be applied in a variety of settings.
4. Innovative: Knowledge engineering is an innovative field that is constantly evolving. Studying knowledge engineering can provide you with the opportunity to work on cutting-edge projects and to contribute to the development of new technologies that have the potential to change the world.
5. Career Growth: Knowledge engineering can offer excellent career growth opportunities, particularly for those who are interested in pursuing a career in data science or artificial intelligence. As the demand for knowledge engineering professionals continues to grow, the salaries and job prospects for those with expertise in this field are likely to remain strong.

Why Machine Learning now?

- Recent progress in algorithms and theory
- Growing flood of (online) data
- Computational power is available

Applications for Machine Learning:

- Data mining: using historical data to improve decisions
  - medical records → medical knowledge
  - customer segmentation: find groups of customers with similar interests
- Software applications we can't program by hand
  - autonomous driving
  - speech recognition: Alexa, Siri, …
  - image recognition: recognizing unkown people on photos
- Self customizing programs
  - newsreader that learns user interests

Combining Knowledge and Machine Learning: Self-driving Cars

- Machine Learning:
  Driving Behaviour

- Knowledge Engineering:
  Traffic Rules

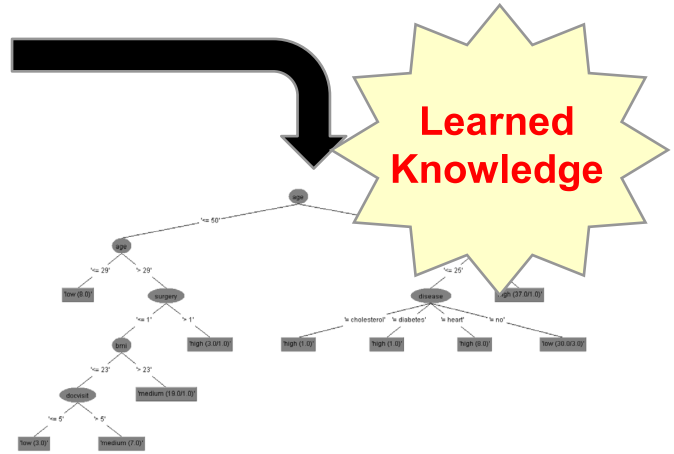
Combining Machine Learning and Knowledge Engineering: Health Insurance (1/3)

Accept yes/no

Combining Machine Learning and Knowledge Engineering (2/3)

- Example: Application of health insurance
  - Machine Learning: data records about risks of clients

Age	surgery	docvisit	allergy	med	disieases	bmi	class
20	0	2	no	no	cholesterol	28	low
21	0	4	no	no	no	23	low
49	2	12	yes	yes	heart	34	high
22	0	3	no	no	no	23	low
51	2	2	yes	yes	diabetes	26	high
52	2	8	no	no	heart	31	high
52	0	3	yes	no	no	22	low
52	2	12	yes	yes	diabetes	27	high
52	0	11	yes	no	cholesterol	29	high
23	0	3	no	no	no	23	low



  - Engineered knowledge: eligibility and compliance

Applicants from Switzerland are eligible.
A person younger than 21 year is not able to apply
…

Engineered Knowledge

Combining Machine Learning and Knowledge Engineering (3/3)

Examples of learned rules:

```
risk (Person, high) :-    age(Person,A), A > 50,
                          bmi(Person, Bmi), Bmi =<25,
                          disease(Person, diabetes).
risk (Person, low) :-     age(Person,A), A =< 29.
```

Learned Knowledge

Examples of engineered rules:

```
eligible(Person, no) :-   age(Person,A), A =< 21.
eligible(Person,no) :-    country(Person,C), C =/= switzerland.
```

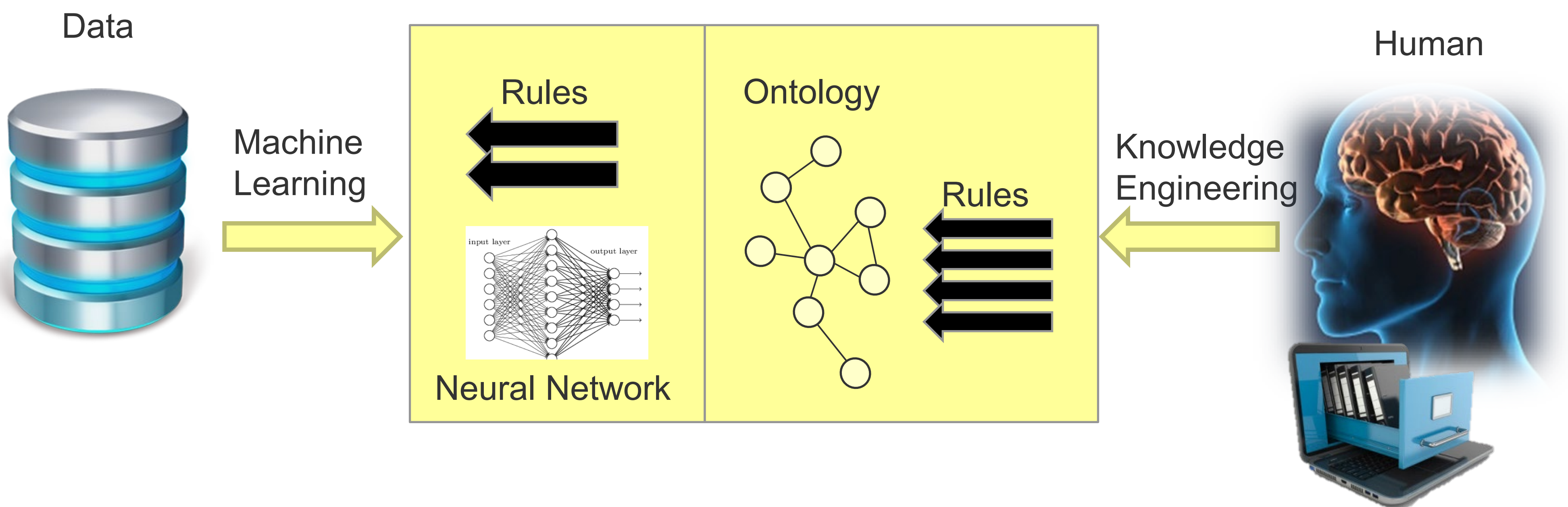
Engineered Knowledge

Combining engineered and learned rules:

```
accept(Person, yes) :- eligible(Person, yes), risk(Person, low).
accept(Person, yes) :- eligible(Person, yes), risk(Person, medium).
accept(Person, no) :- eligible(Person, no).
accept(Person, no) :- risk(Person, high)
```

Engineered Knowledge

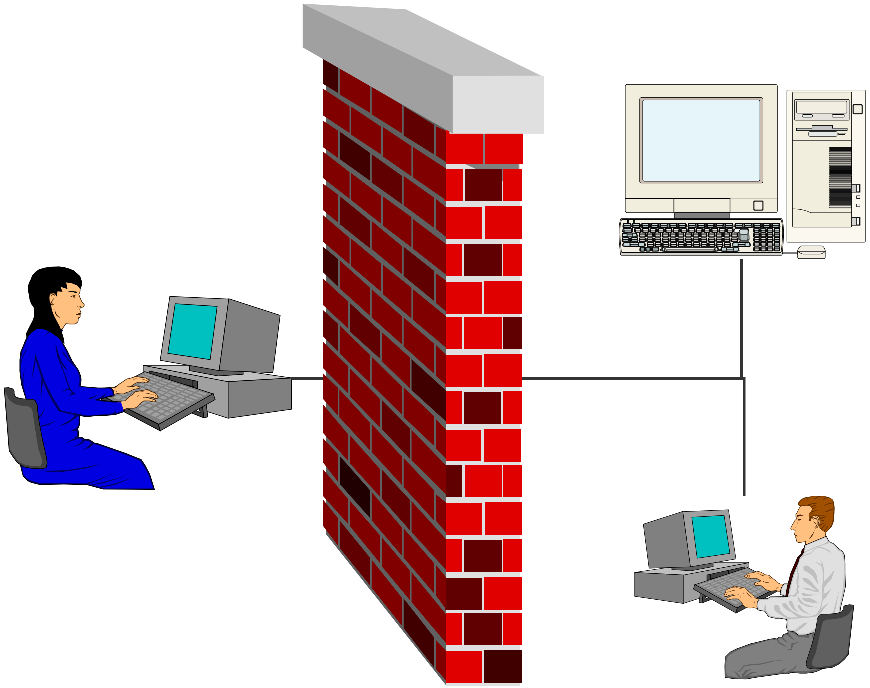
Summary:
Knowledge Sources in a Knowledge Base



tacit knowledge &
new knowledge from data

self-aware knowledge &
external knowledge

Are Machines Able to Think? – The Turing-Test



- Are Machines able to think?
- In order to find an answer to this question, the English computer pioneer A. Turing developed 1950 the so-called Turing-Test
- Test arrangement:
  - Room A: interviewer
  - Room B: Computer and Human
- The interviewer asks questions from different fields aiming to discover whether the computer or the human has provided the answer.
- The computer has passed the Turing-Test, if the interviewer cannot say who answers the questions, the computer or the human.