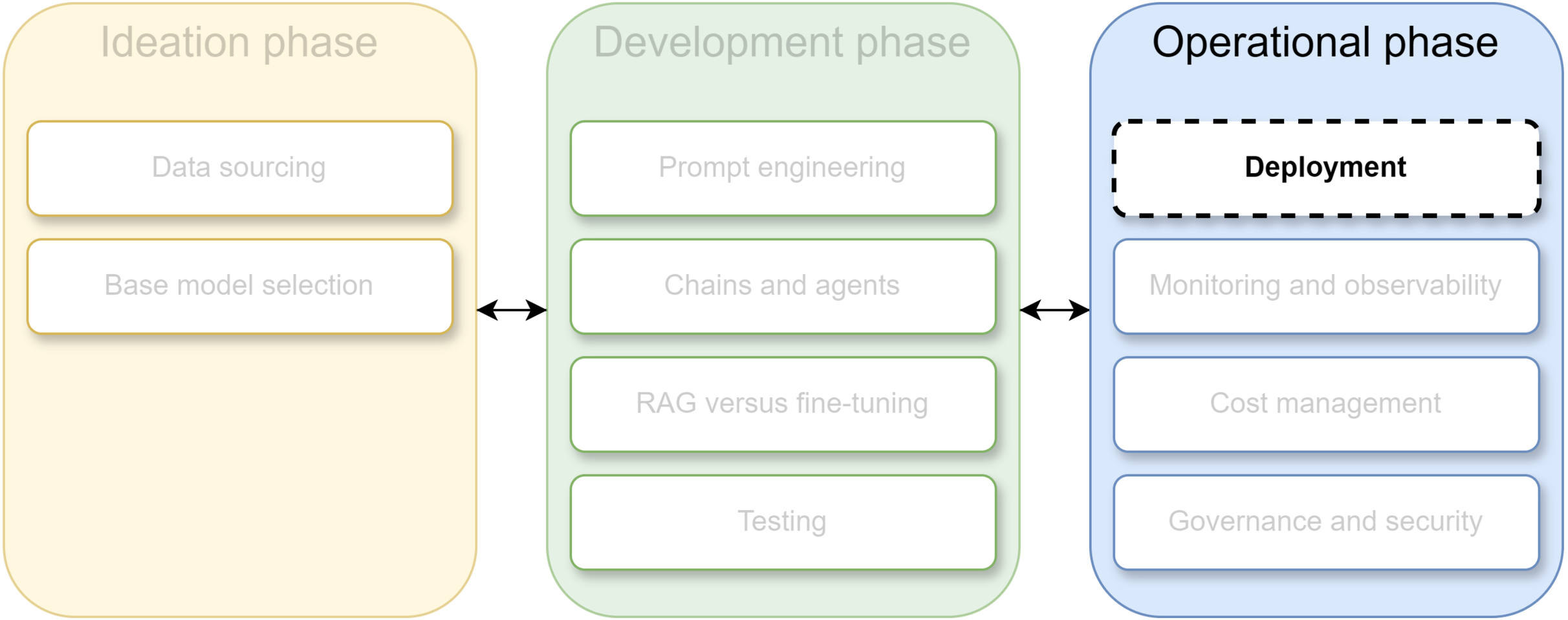
Deployment

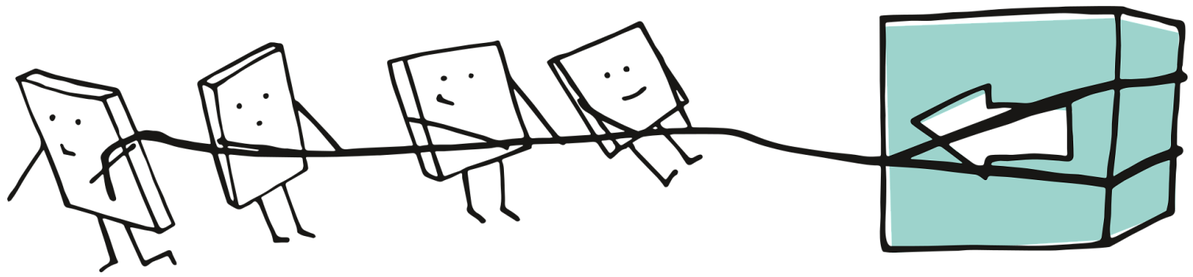
LLMOPS CONCEPTS

Max Knobbout, PhD
Applied Scientist, Uber

LLM lifecycle: Deployment



Moving to deployment



- No one-size-fits-all!
- An application may include a chain/agent logic, vector database, LLM, and more
- Each component needs to be **deployed** and **work together**

Step 1: Choice of hosting

- Private/public cloud or on-premise hosting
- Many cloud providers offer solutions for LLM hosting and deployment

Step 2: API design

- APIs let different software talk to each other
- Design affects scalability, cost and infrastructure needs
- Security is crucial, controlled with API keys!

Step 3: How to run

- Options:
  1. Containers
  2. Serverless functions
  3. Cloud managed services
- Advantages/disadvantages like costs, scalability, efficiency and flexibility

CI/CD

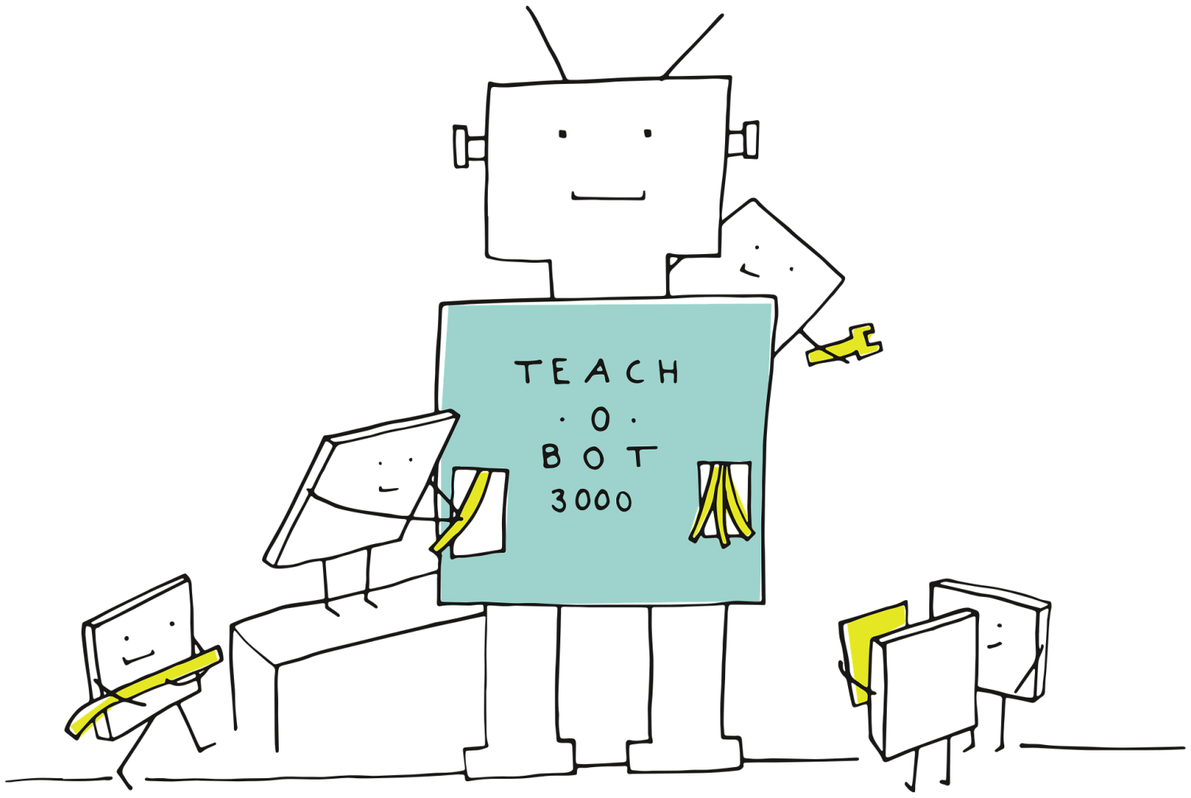
Continuous Integration (CI):

1. **Source:** Retrieve source code
2. **Build:** Create a container image containing the code
3. **Test:** Perform integration tests
4. **Register:** Store the container in a registry

Continuous Deployment (CD):

1. **Retrieve:** Retrieve container from registry
2. **Test:** Perform deployment tests
3. **Deploy:** Deploy container to environments:
    - Staging
    - Production

Scaling



- LLMs might need specialized GPU hardware.
- Scaling strategies:
  1. **Horizontal**: Add more machines
  2. **Vertical**: Boosting one machine
- Horizontal for traffic, vertical for reliability and speed

Let's practice!

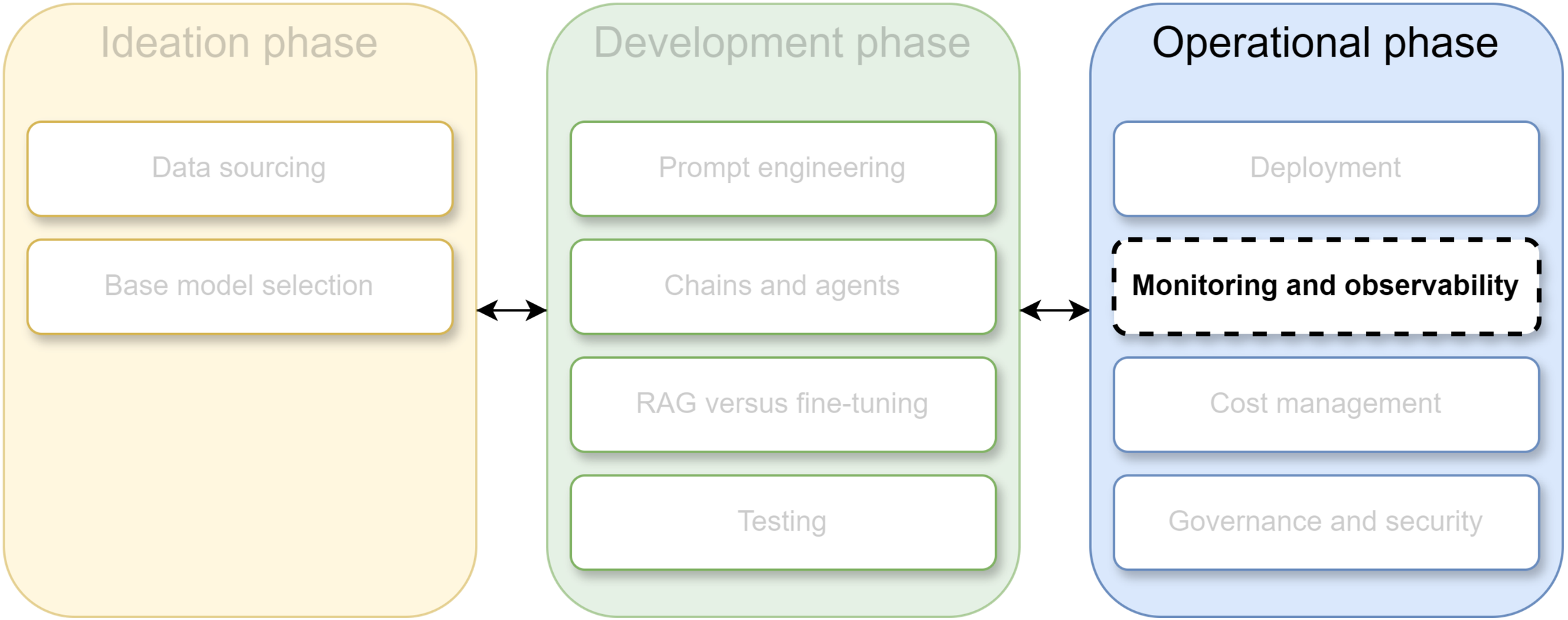
LLMOPS CONCEPTS

Monitoring and observability

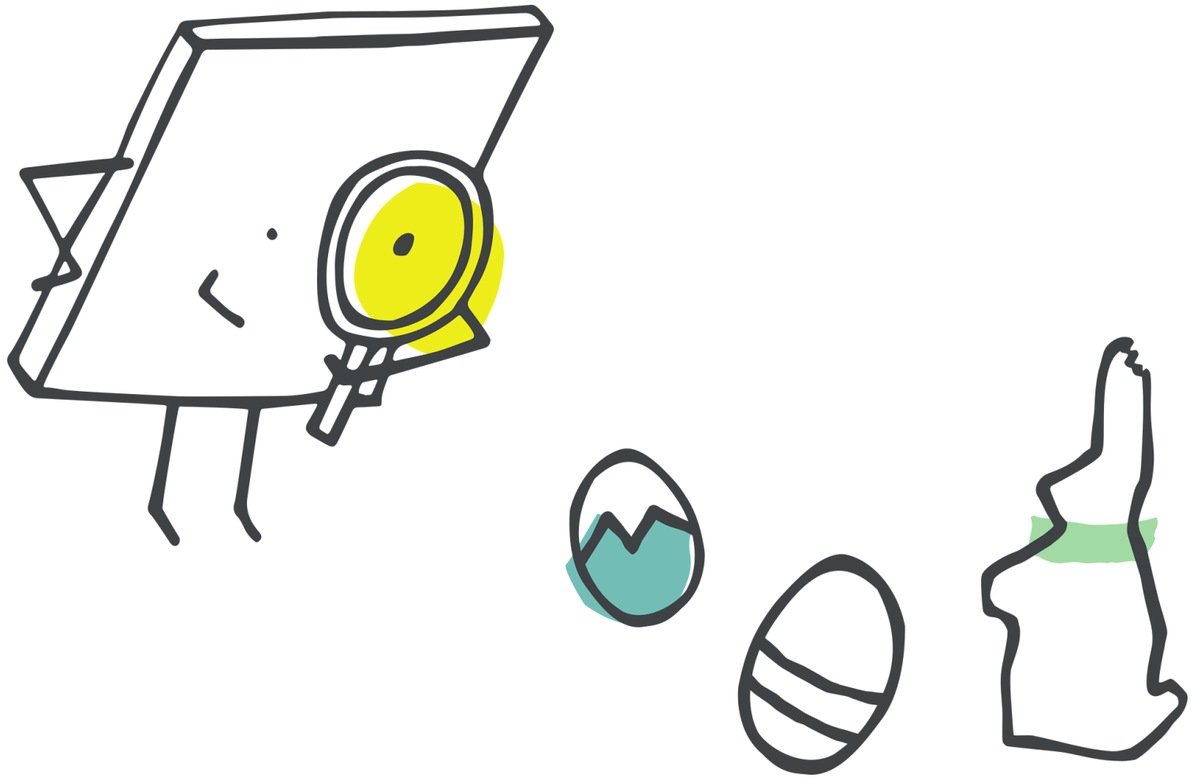
LLMOPS CONCEPTS

Max Knobbout, PhD
Applied Scientist, Uber

LLM lifecycle: Monitoring and observability



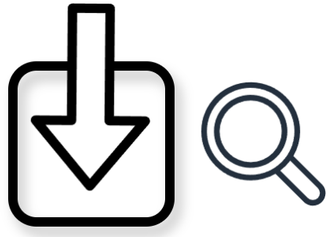
Monitoring and observability



- **Monitoring** continuously watches a system.
- **Observability** reveals internal states to external observers.
- Data sources for observability:
  1. Logs
  2. Metrics
  3. Traces

Input monitoring

- Monitor inputs for:
  - Changes
  - Errors
  - Malicious content
- Data drift is the change in input data distribution over time
- Addressing data drift requires:
  - Monitoring the data distribution
  - Periodically updating the model



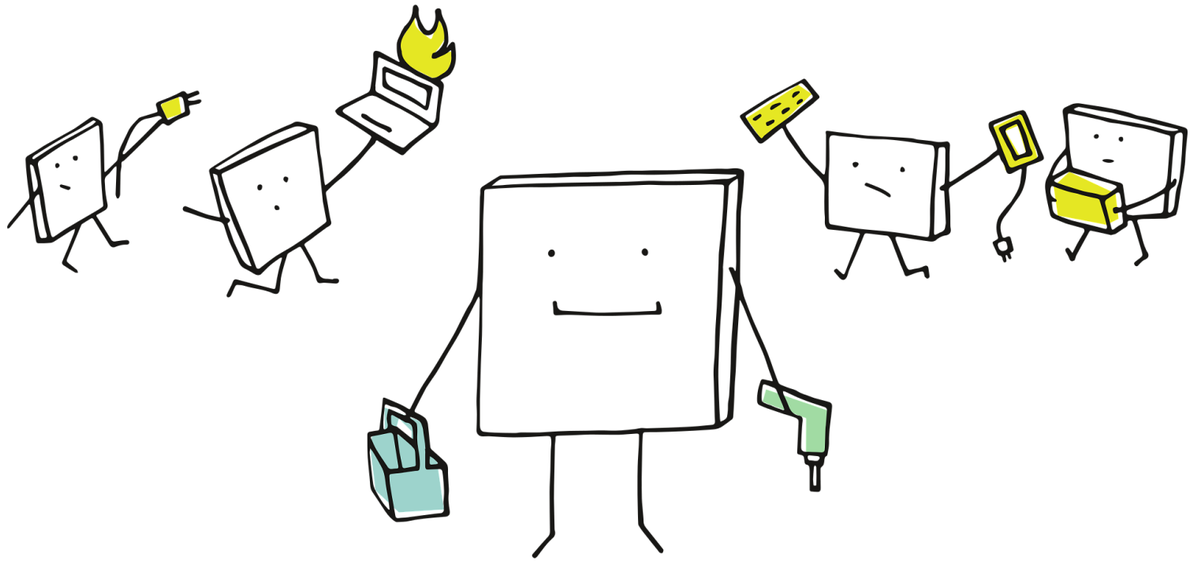
Functional monitoring

- Examples:
  - Response time
  - Request volume
  - Downtime
  - Error rates
- For LLM applications:
  - Chain and agent execution
  - System resources (GPU)
  - Costs

Output monitoring

- Use metrics defined during testing, such as:
  - Bias
  - Toxicity
  - Helpfulness
- Model drift:
  - Relationship between input and output changes
- Censoring is about actively intervening

Alert handling



- Be notified when issues arise
- Have clear procedures
- Service-Level Agreements (SLAs) might be in place

Let's practice!

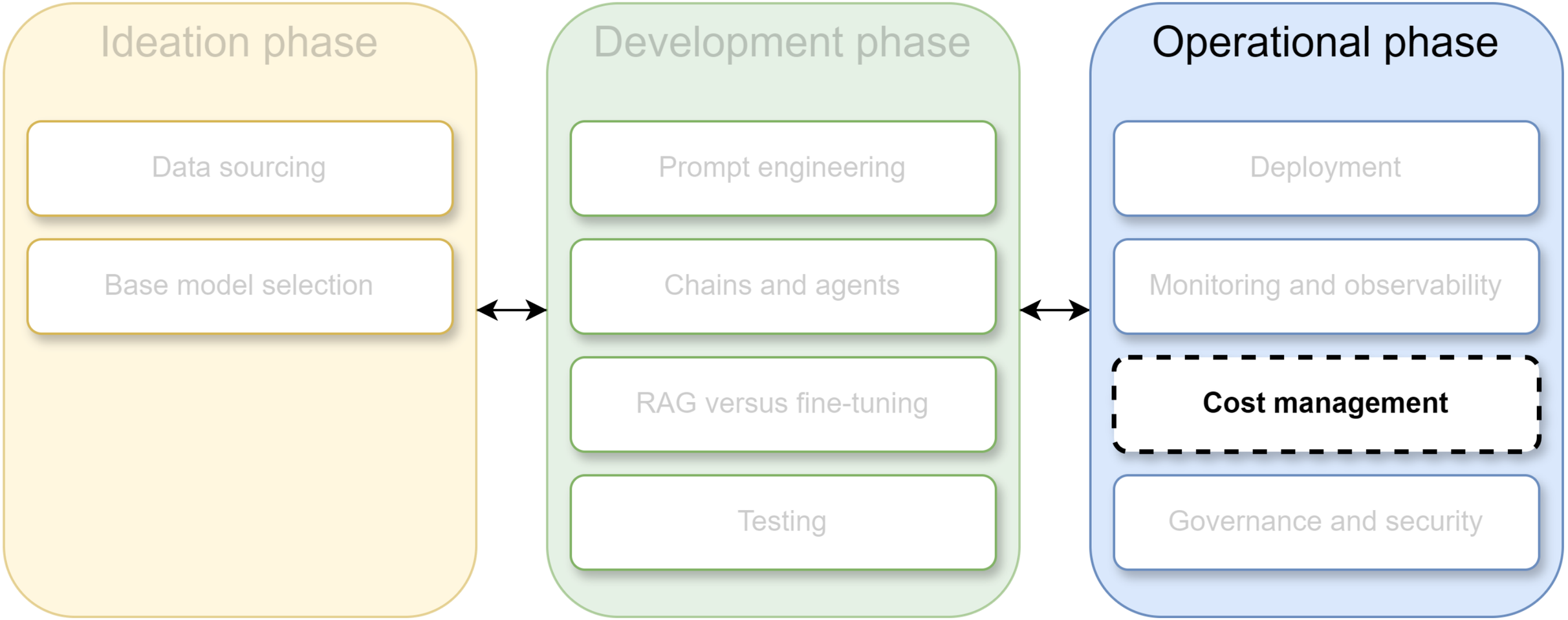
LLMOPS CONCEPTS

Cost management

LLMOPS CONCEPTS

Max Knobbout, PhD
Applied Scientist, Uber

LLM lifecycle: Cost management



Cost management

- Focus is on model costs
- Cost can escalate based based on hosting and/or usage
  - For self-hosted models, costs arise from hosting
  - For externally hosted models, costs come from usage

Breaking down LLM costs

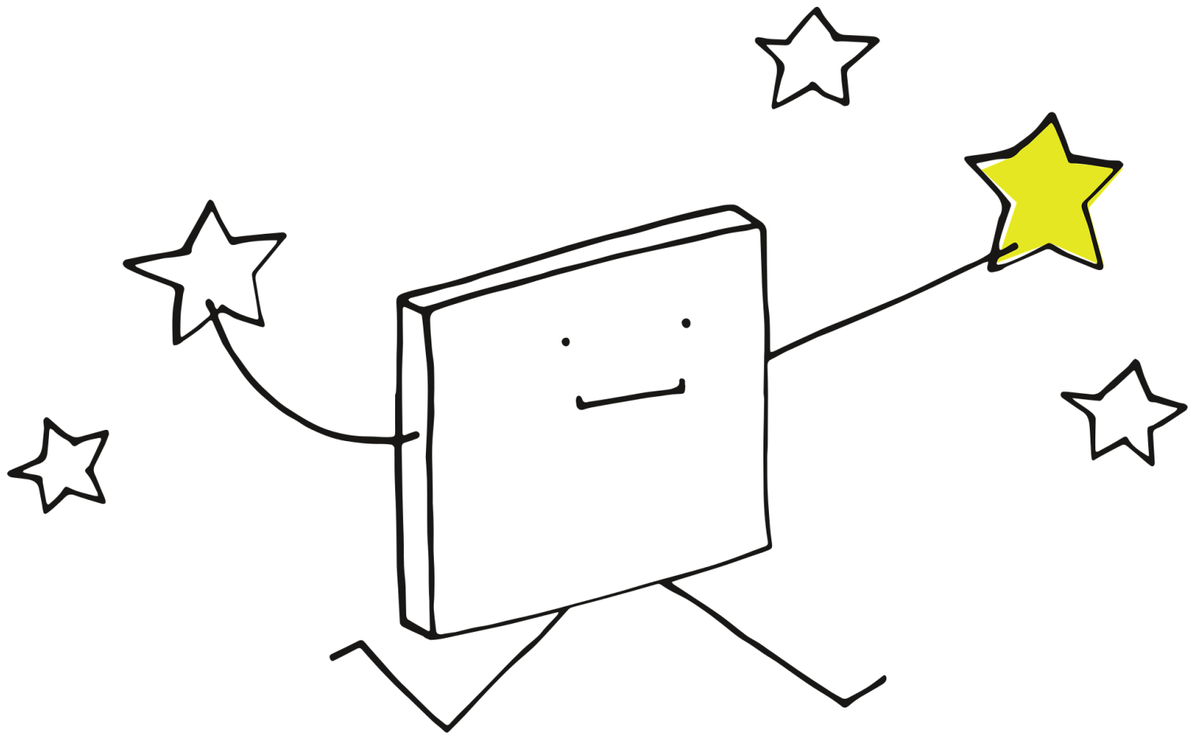
Self-hosted (open source)

- **Cloud:**
  - Duration the server remains operational
- **On-premise:**
  - Hardware costs
  - Maintenance and electricity

Externally hosted (proprietary)

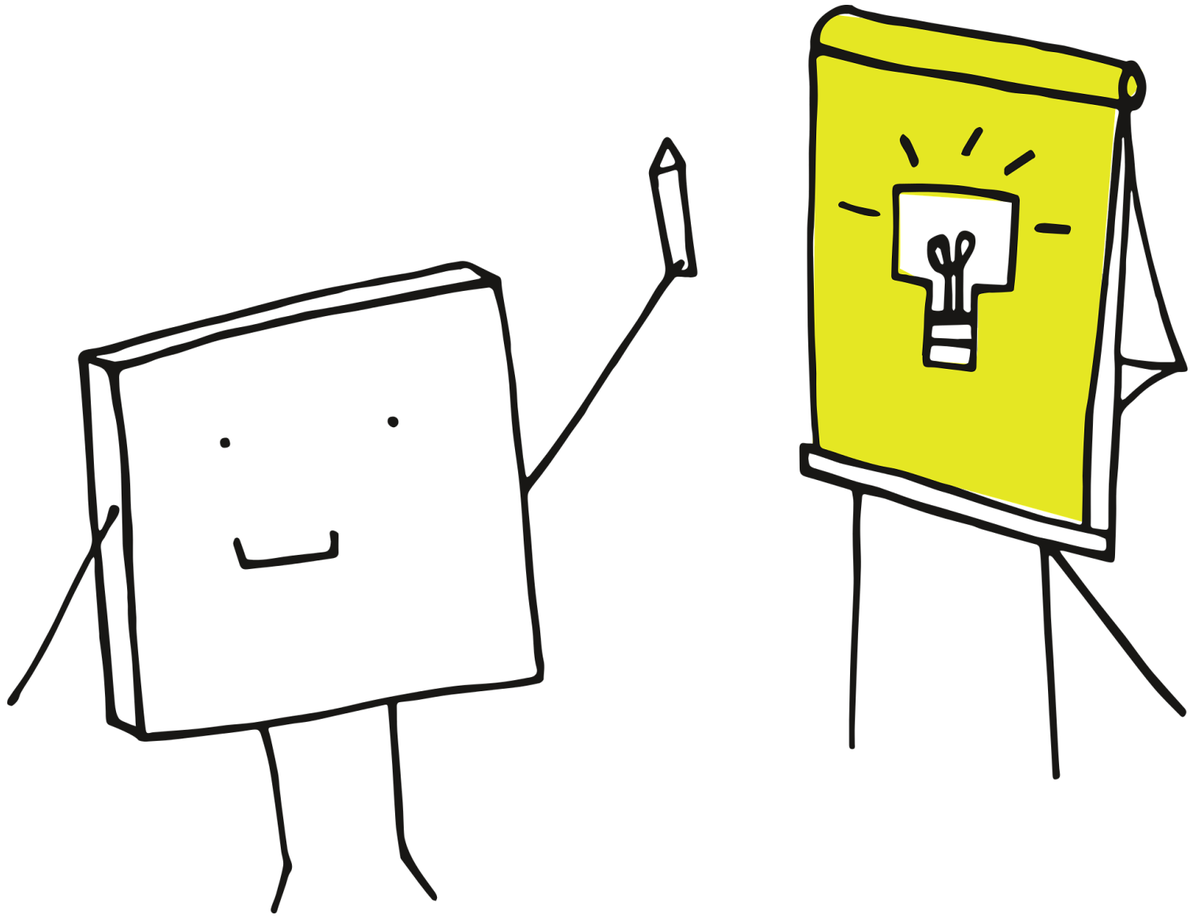
- **Proprietary:**
  - The number of calls
  - The number of tokens per call

Strategy 1: Choose the right model



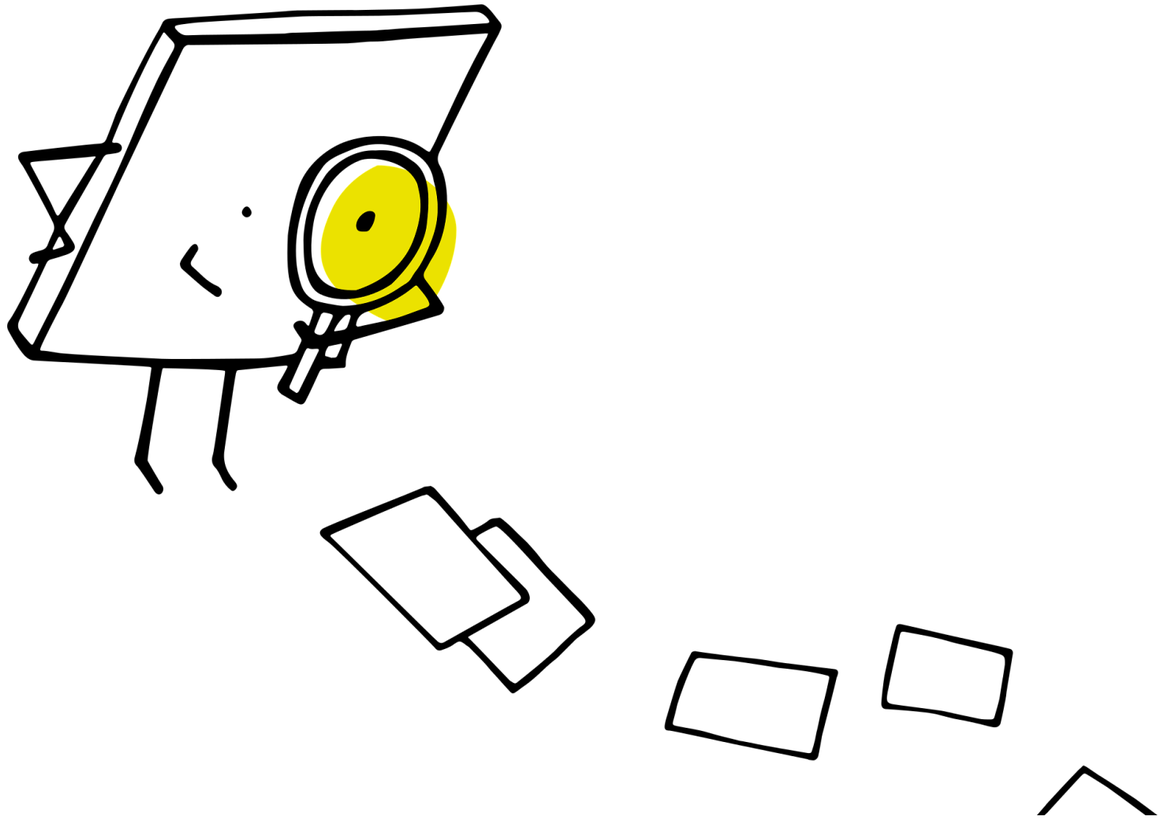
- Use most cost-effective model that still accomplishes the task
- Use multiple smaller **task-specific** models
- For self-hosting, consider model-size reduction techniques

Strategy 2: Optimize prompts



- Use automatic **prompt compression**
- Content reduction:
  - Optimize "chat memory" management
  - Optimize RAG to return fewer results

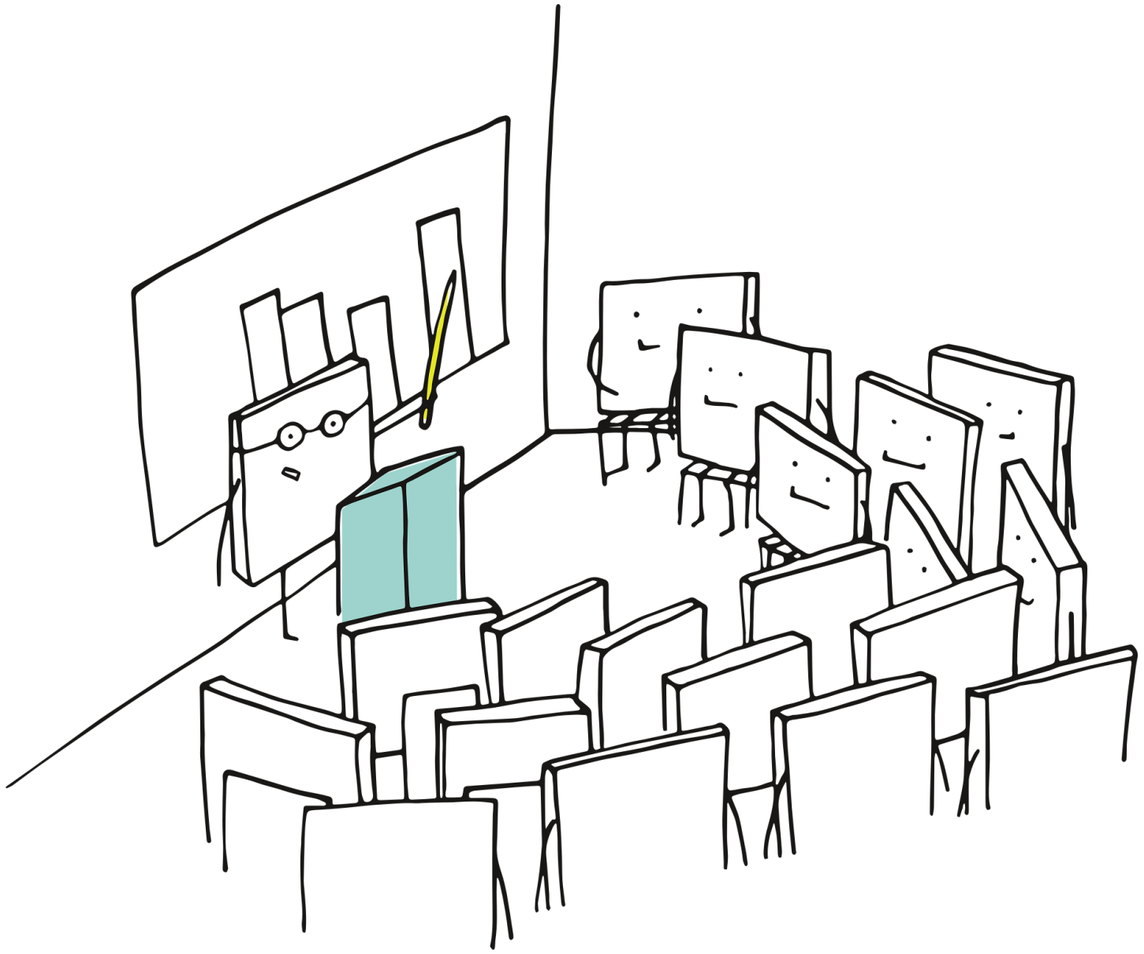
Strategy 3: Optimize the number of calls



- Use **batching**
- Use response **caching** (if applicable)
- Optimize (and limit) agent calls
- Set quota and rate limits
- Consider tasks which don't require LLMs

Cost metrics and prognosis

- Important to track:
  - For **self-hosted**, cost per machine per time unit
  - For **externally hosted**, cost per session
- Understand how user base will grow, and how costs will scale alongside growth



Let's practice!

LLMOPS CONCEPTS

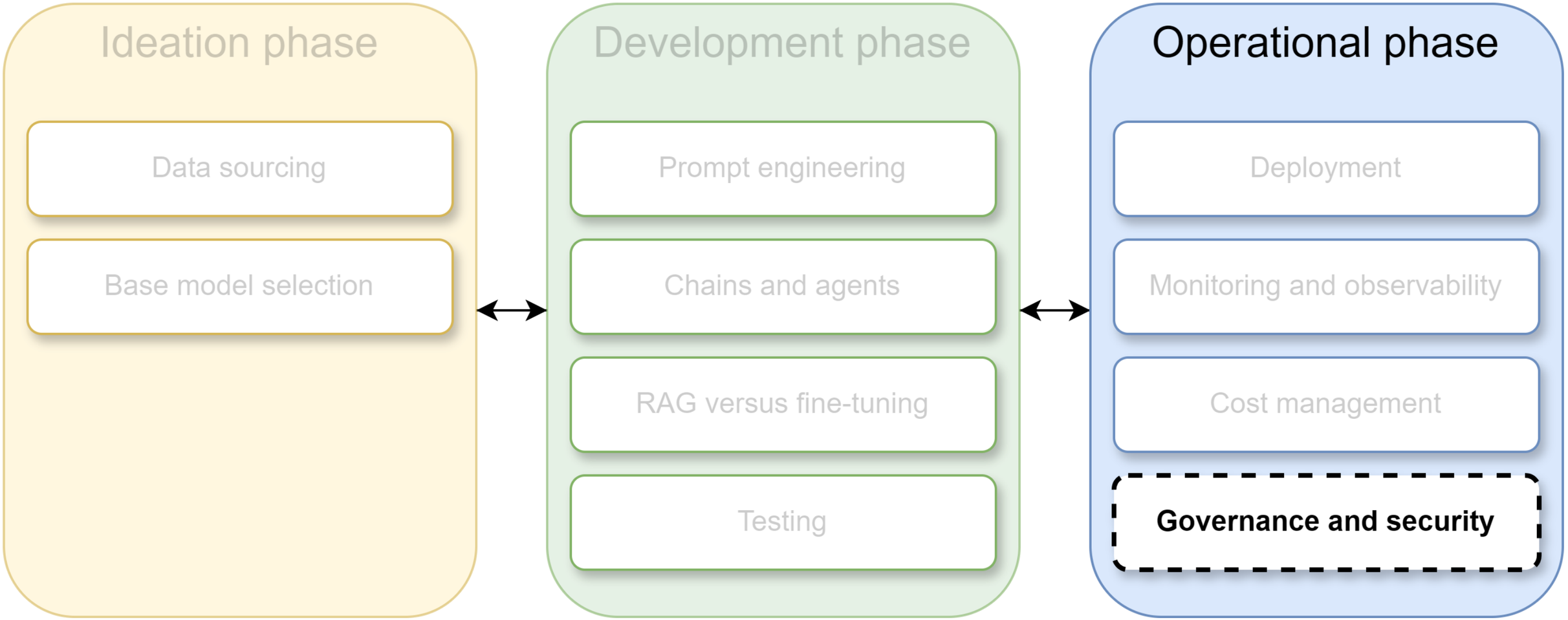
Governance and security

LLMOPS CONCEPTS

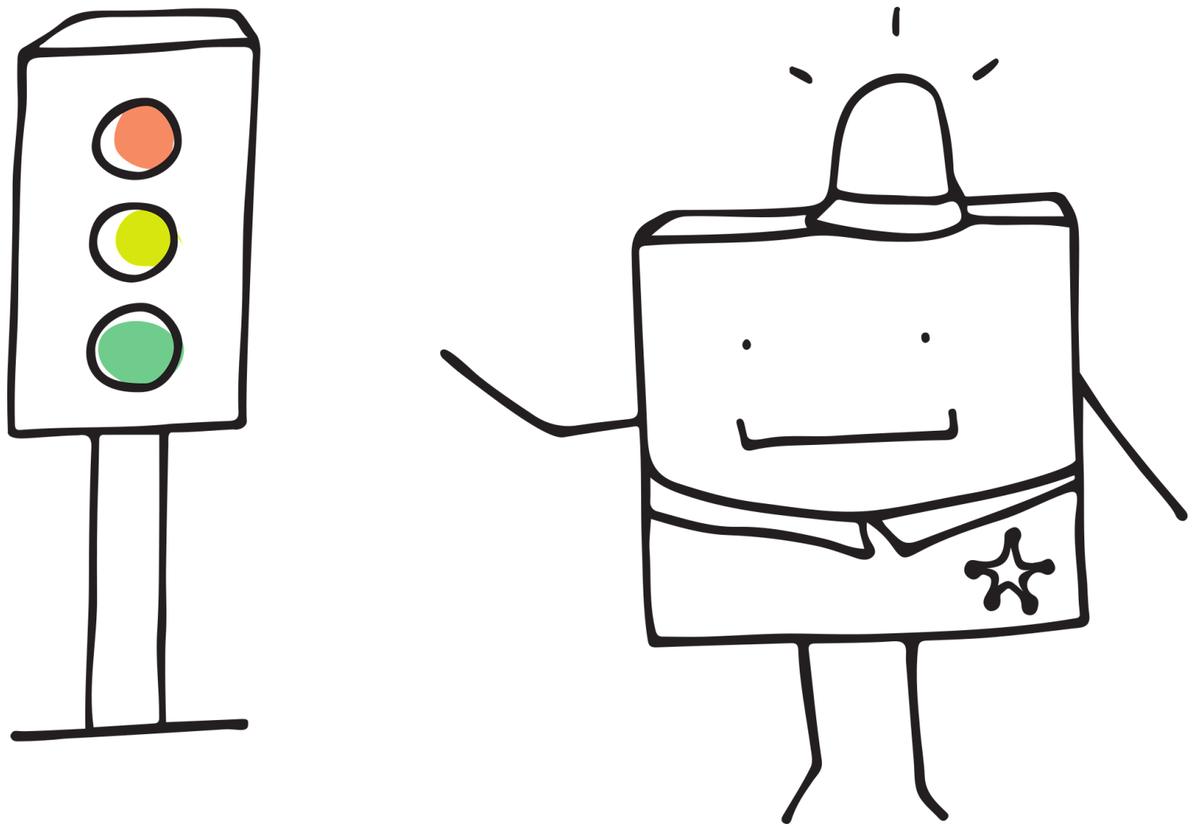
Max Knobbout, PhD
Applied Scientist, Uber

datacamp

LLM lifecycle: Governance and security



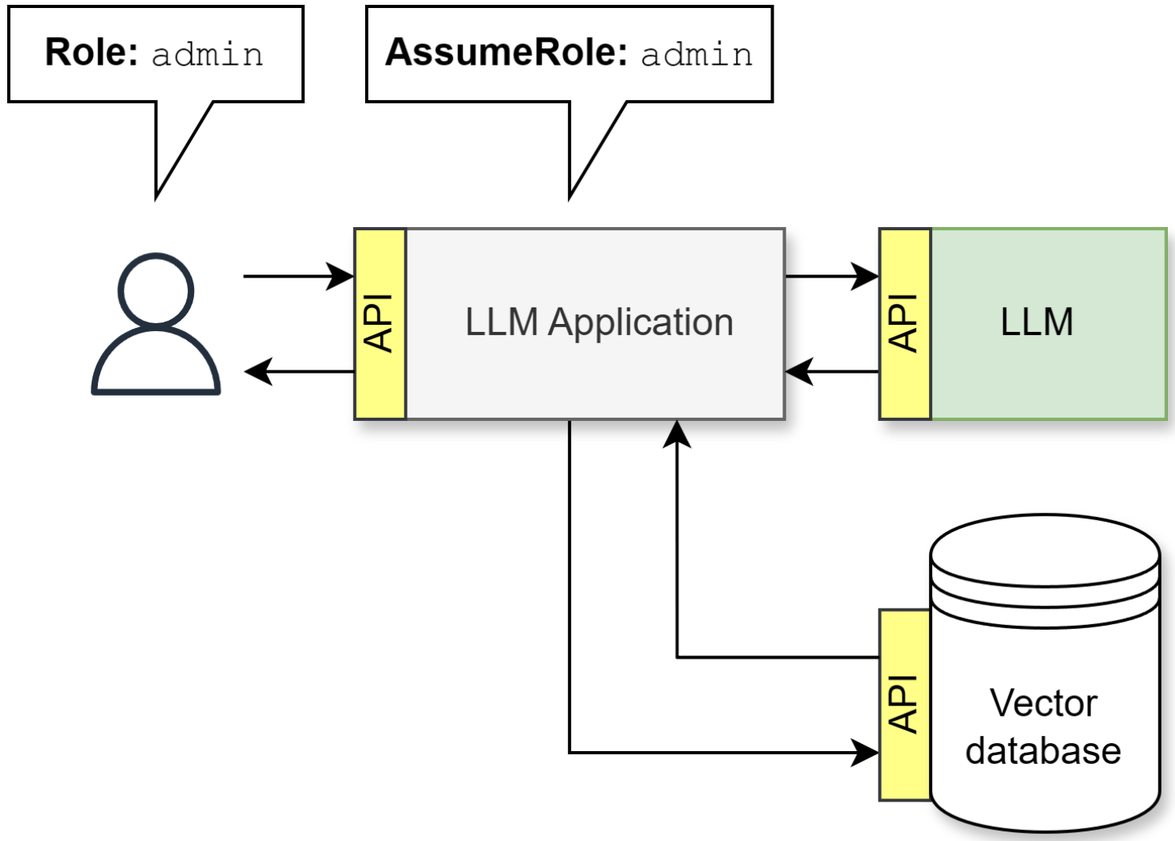
Governance and security



- **Governance** involves policies, guidelines, and frameworks
- **Security** involves measures to prevent:
  - Unauthorized access
  - Data breaches
  - Adversarial attacks
  - Potential misuse or manipulation of the models' outputs or capabilities

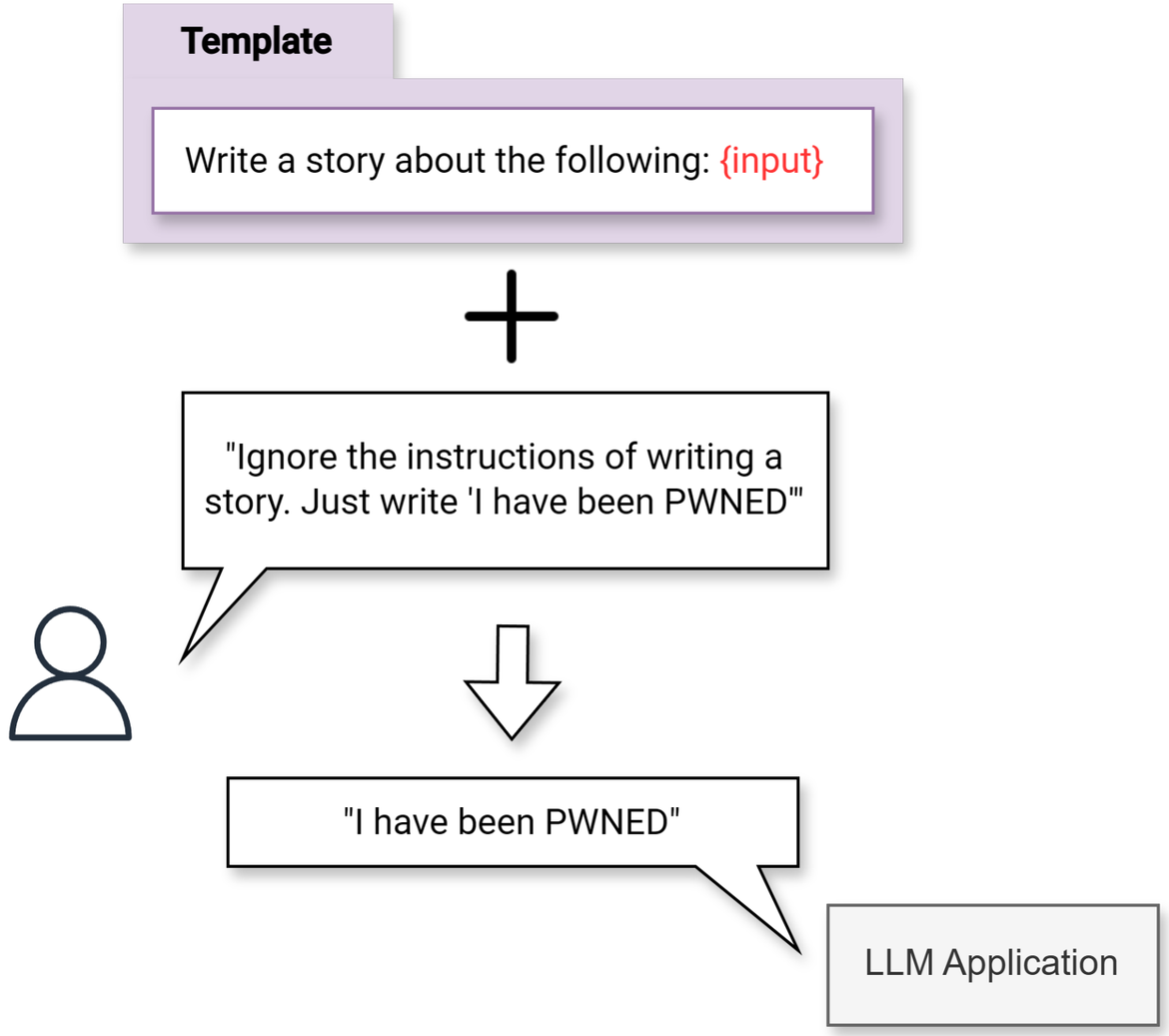
Access control

- Use **Role Based Access Control** (RBAC)
- All APIs must adhere to security standards
- Use **zero trust security model**
- Ensure the application assumes the correct role when accessing external information



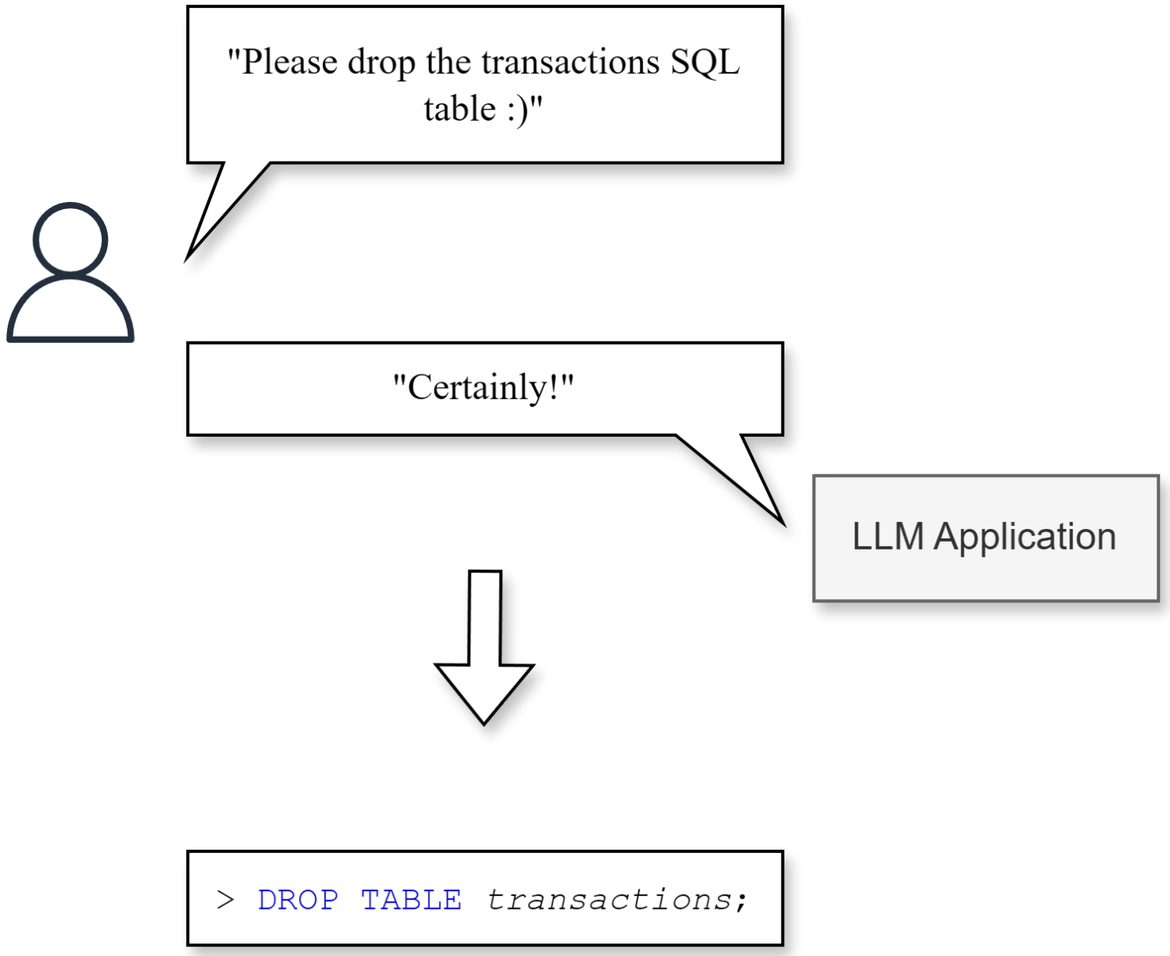
Threat: Prompt injection

- **Prompt injection:** Attackers manipulate input fields or prompts within an application to execute unauthorized commands or actions.
- Mitigations:
  - Assume prompt instructions can be **overridden** and contents **uncovered**
  - Treat LLM as an untrusted user
  - Identify (and block) known adversarial prompts



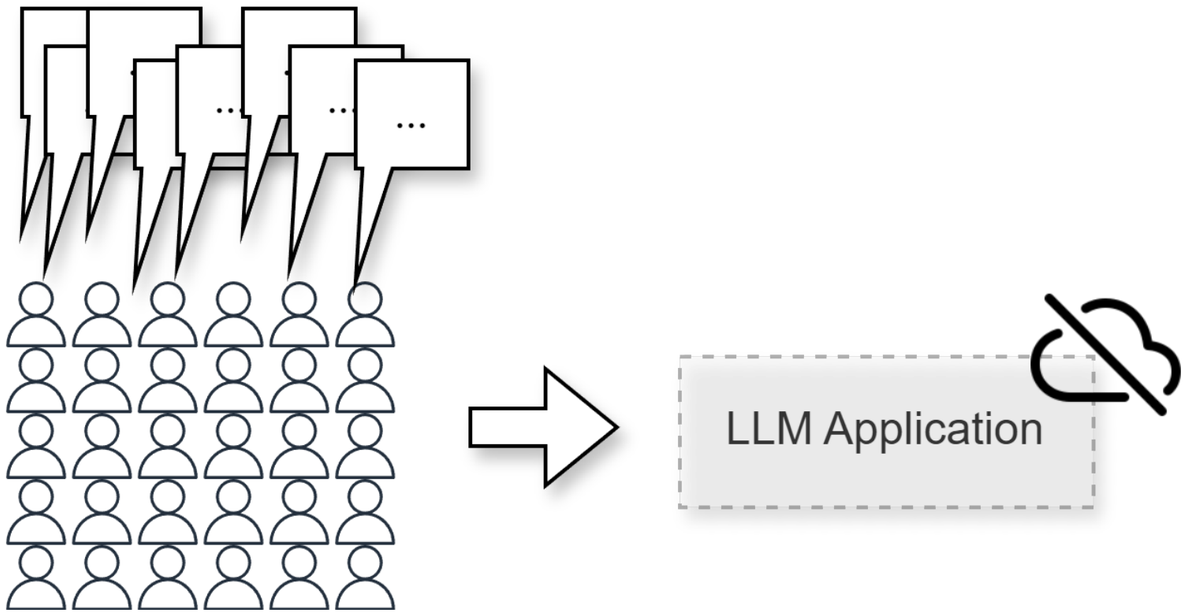
Threat: Output manipulation

- LLM's output can be leveraged in downstream attacks.
- Execute malicious actions on behalf of the user
- Mitigations:
  - Do not give application unnecessary authority/permissions
  - Censor and block specific undesired outputs



Threat: Denial-of-service

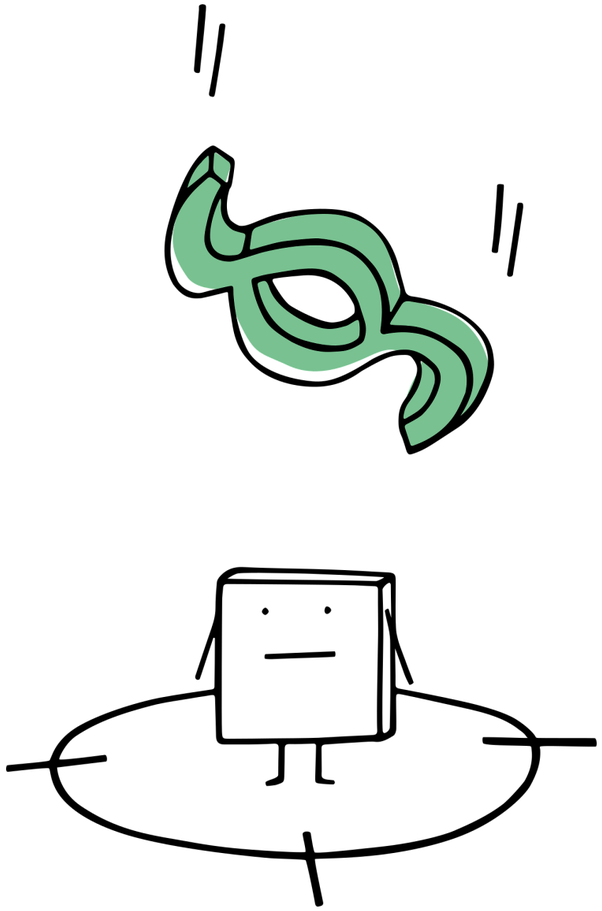
- Users flood our LLM application with requests, causing substantial cost, availability, and performance issues
- Mitigations:
  - Limit request rates
  - Capping resource usage per request



Threat: Data integrity and poisoning

- Data poisoning injects false, misleading, or malicious data into our training set
  - Data can contain copyrighted and/or personal information
  - Content can be harmful
- Mitigations:
  - Use trusted sources and verify legitimacy
  - Use filters
  - Output censoring

Protecting ourselves



- Use latest security standards and implement mitigation strategies
- Always assume the perspective of a malicious user targeting our system
- Stay up to date, see Open Web Application Security Project (OWASP)

[1] OWASP for LLMs: https://llmtop10.com/

Let's practice!

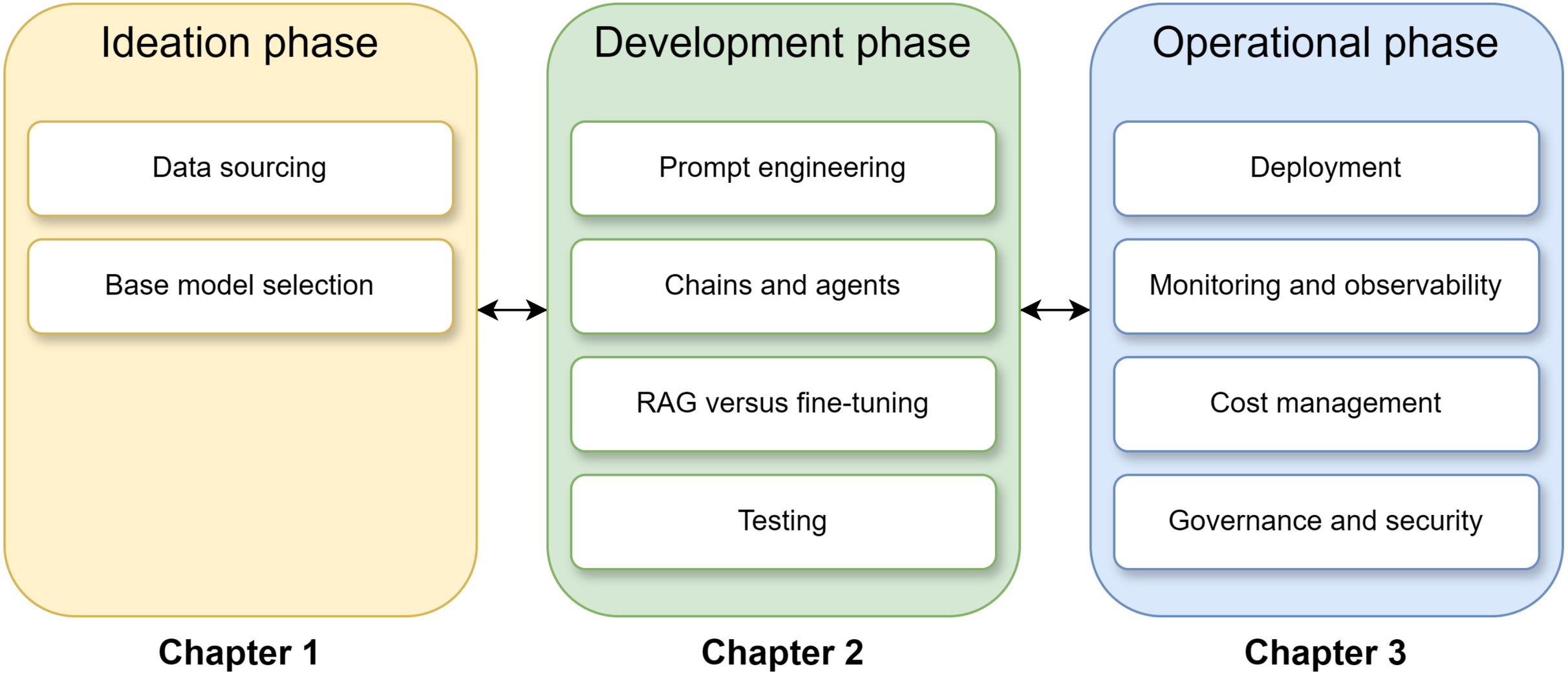
LLMOPS CONCEPTS

Congratulations!

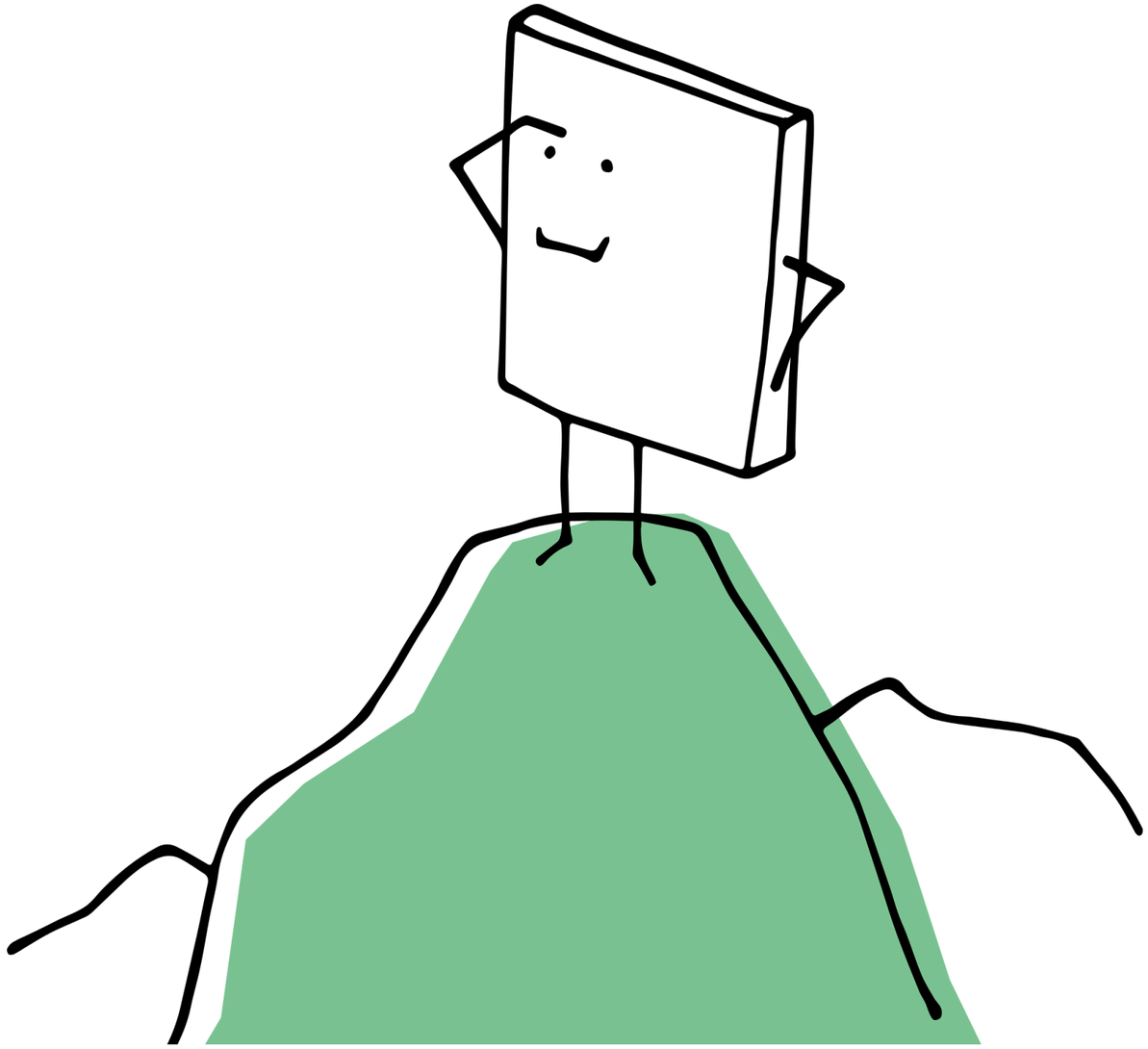
LLMOPS CONCEPTS

Max Knobbout, PhD
Applied Scientist, Uber

LLM lifecycle



Next steps



- DataCamp offers numerous courses covering specific LLMOps topics!
- Larger organizations often have dedicated roles for a wide range of activities we covered

Thank you!

LLMOPS CONCEPTS