

# OP Pohjola Observability and Automation

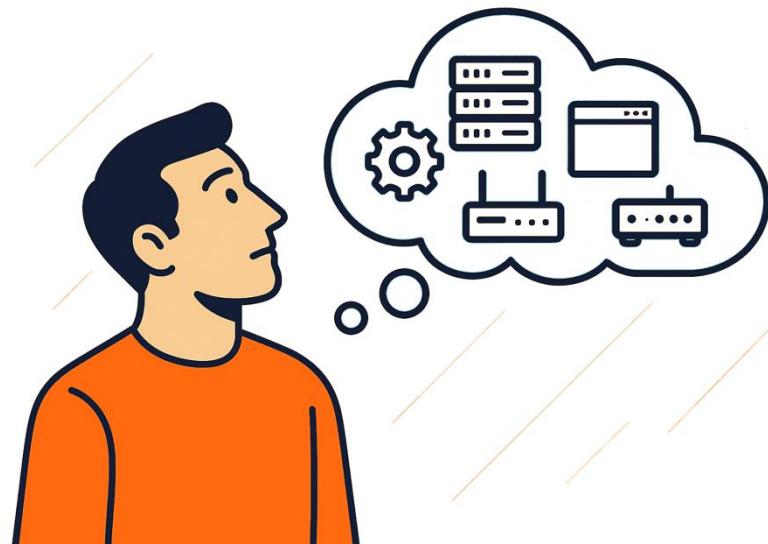
Past, Present and Future

# Compliance Disclaimer

- OP Pohjola is the largest cooperative bank in Finland
- Information presented should not be considered official statements by OP Pohjola
- Official information from proper channels supersedes information mentioned in these slides
- Content of this presentation does not represent official policy, but technical insights, experiences and personal opinions

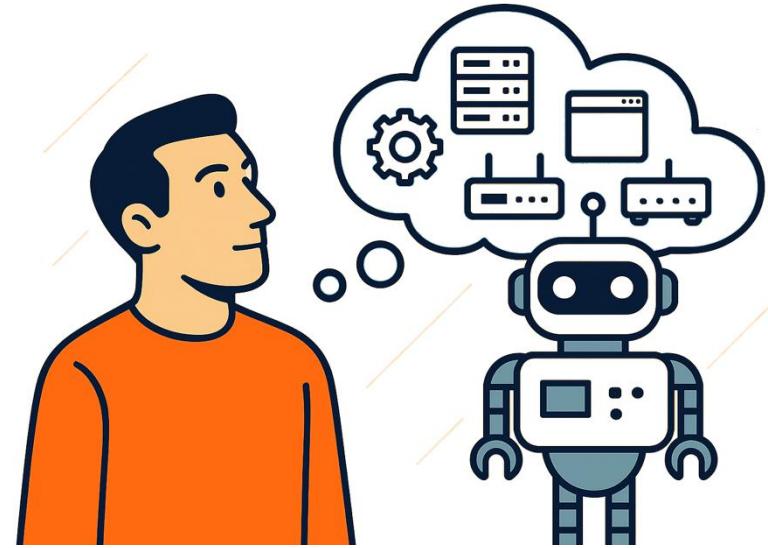
# World Before Observability

- Monitoring only collects metrics
- Manual correlation of metrics
- Semi-manual log analytics
- Limited automation



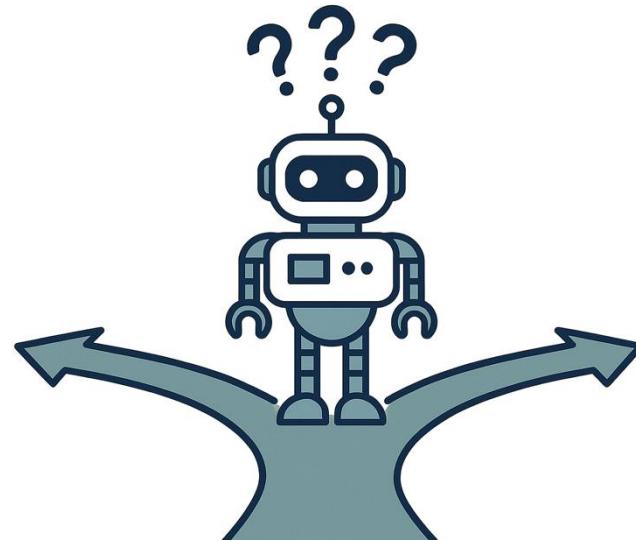
# Initial Dynatrace Adoption (AppMon)

- Team-sized testing in non-prod
- Discovery of issues in test env.
- Traces are a feature you can count on
- Dashboards and better UX
- Easily manageable instrumentation / auto instrumentation
- Learning about observability (not just monitoring – understanding and reacting to events in context)



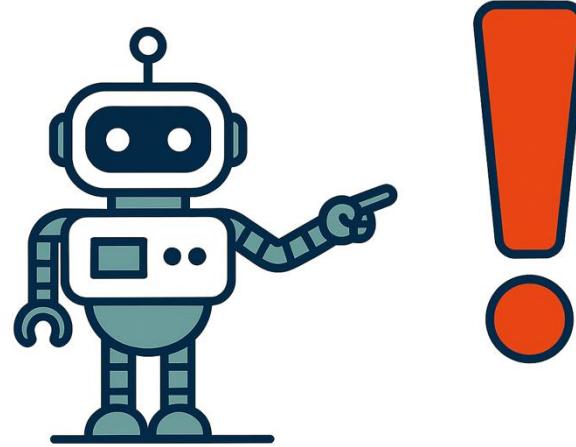
# Early DevOps Integration Experience

- Trainings and initial interest
- Tried implementing quality gateways
- Gave up due to multi-faceted nature of our build pipeline



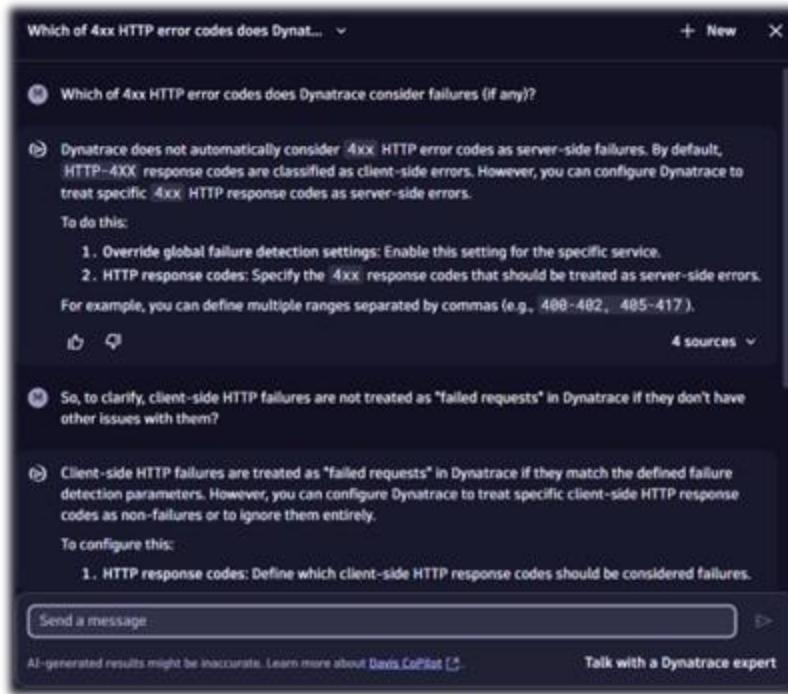
# Davis AI in Production Monitoring

- Better reaction times
- Problems and baselines
- Continuous improvement



# Davis AI CoPilot

- Onboarding new users
- Explaining documentation
- Generating DQL



Which of 4xx HTTP error codes does Dynatrace... ▾ + New X

Which of 4xx HTTP error codes does Dynatrace consider failures (if any)?

Dynatrace does not automatically consider 4xx HTTP error codes as server-side failures. By default, HTTP-4XX response codes are classified as client-side errors. However, you can configure Dynatrace to treat specific 4xx HTTP response codes as server-side errors.

To do this:

1. Override global failure detection settings: Enable this setting for the specific service.
2. HTTP response codes: Specify the 4XX response codes that should be treated as server-side errors.

For example, you can define multiple ranges separated by commas (e.g., 400-402, 405-417).

4 sources ▾

So, to clarify, client-side HTTP failures are not treated as "failed requests" in Dynatrace if they don't have other issues with them?

Client-side HTTP failures are treated as "failed requests" in Dynatrace if they match the defined failure detection parameters. However, you can configure Dynatrace to treat specific client-side HTTP response codes as non-failures or to ignore them entirely.

To configure this:

1. HTTP response codes: Define which client-side HTTP response codes should be considered failures.

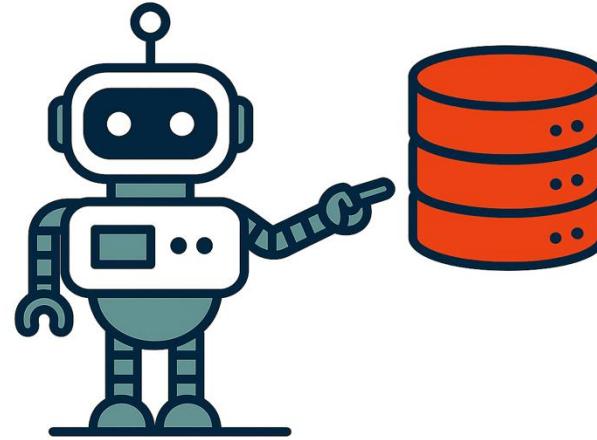
Send a message ▾ ▶

AI-generated results might be inaccurate. Learn more about Davis CoPilot ▾

Talk with a Dynatrace expert

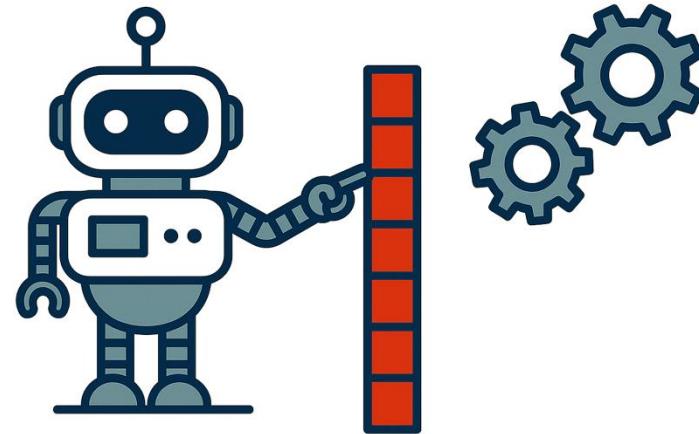
# Grail and DQL

- Grail Data Lakehouse
- DQL in Notebooks and Dashboards
- Querying spans
- Custom request tags



# Workflows and EdgeConnect

- Automate around Dynatrace events and data
- Process data and send notifications
- Execute actions in Kubernetes environments with EdgeConnect

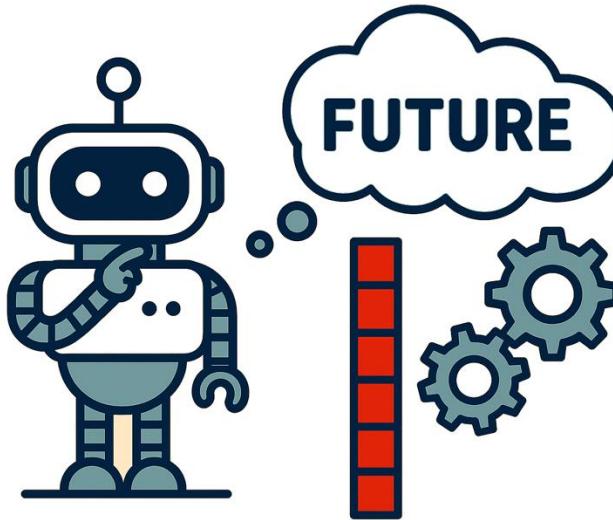


# AI Adoption – Policy and Feeling

- › Customer first
- › Legislative norms define what we can do with AI (f.ex. GDPR and others)
- › Focus on measurable results and tangible ROI
- › Innovation is for everyone and by everyone
- › No AI for AI's sake

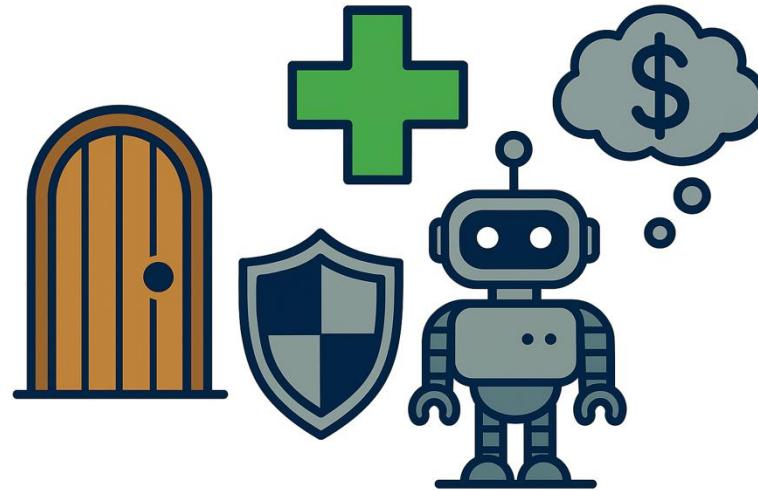
# Future: Predictive Analytics

- Davis Predictive
- Predictive Dashboards
- Predictive Alerts
- Predictive Workflows



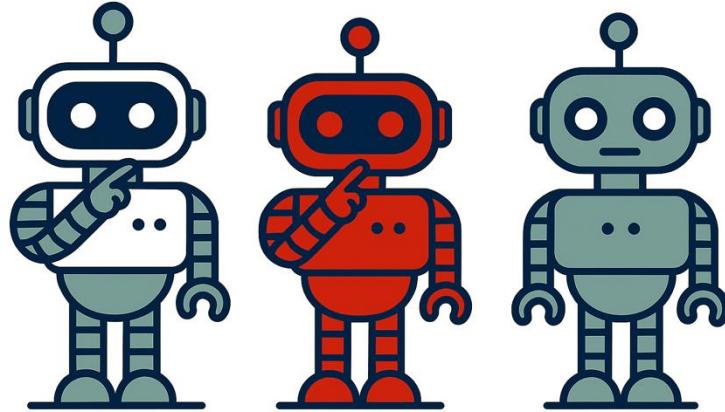
# Future: Autonomous Operations

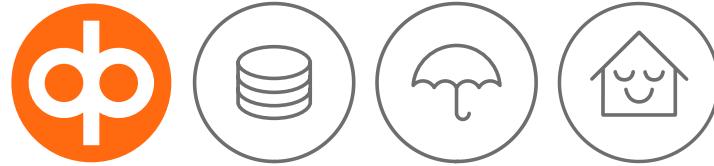
- Quality Gateways
- Automated Security
- Self-Healing Systems (Remediation)
- Automated Incident Response
- Human in the loop? (who is the decision maker)
- FinOps in Cloud



# Future: Innovation & Emerging Technologies

- Managing technology debt with AI
- Agentic Workflows: integrating Dynatrace with ServiceNow and other systems with MCP
- Knowledge Management





# Q & A