AAD / Backend Development

Observability

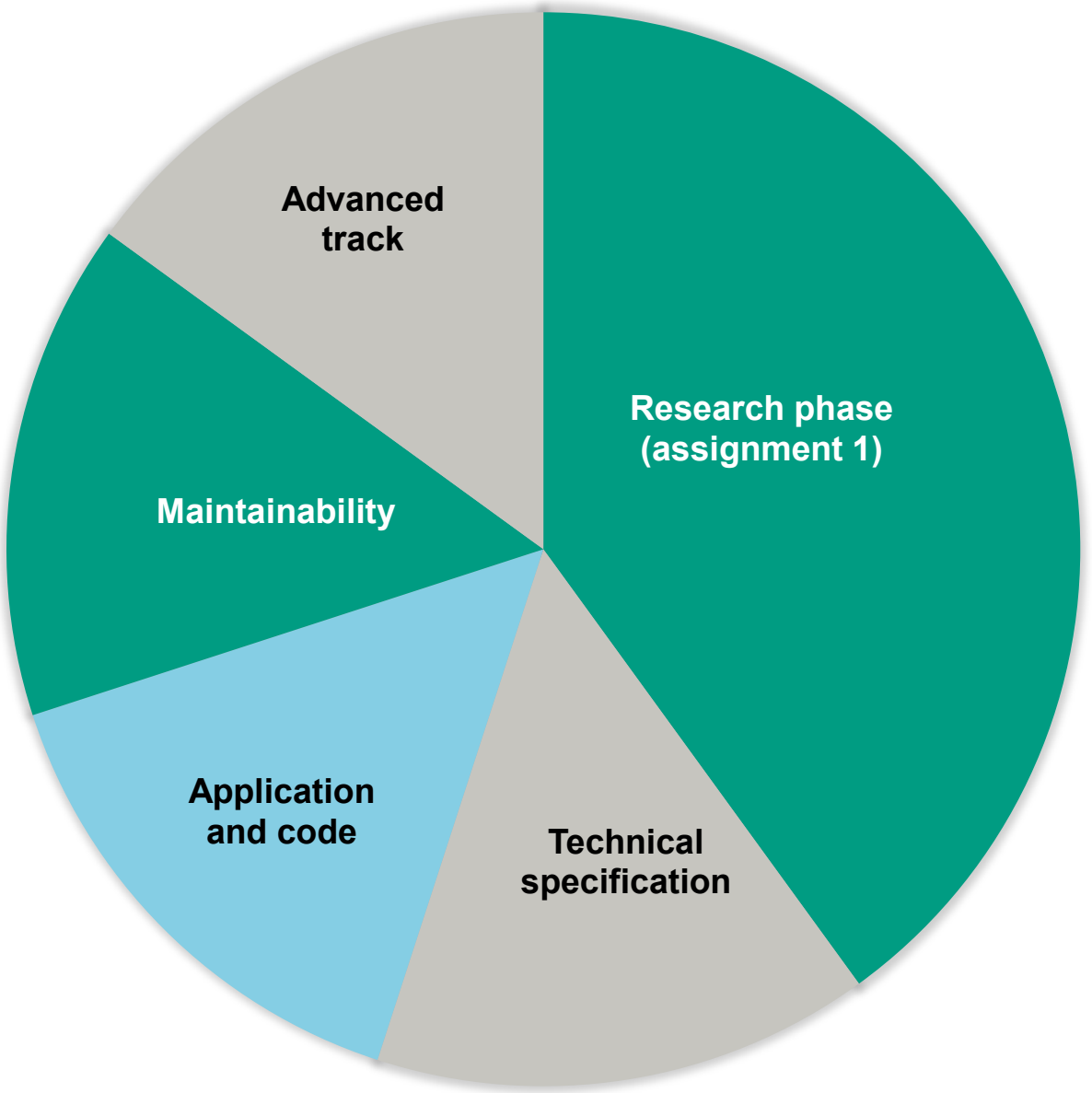
Erik van der Arend / Jan Willem Boer

23/24 Q3

SAXION
UNIVERSITY OF
APPLIED SCIENCES

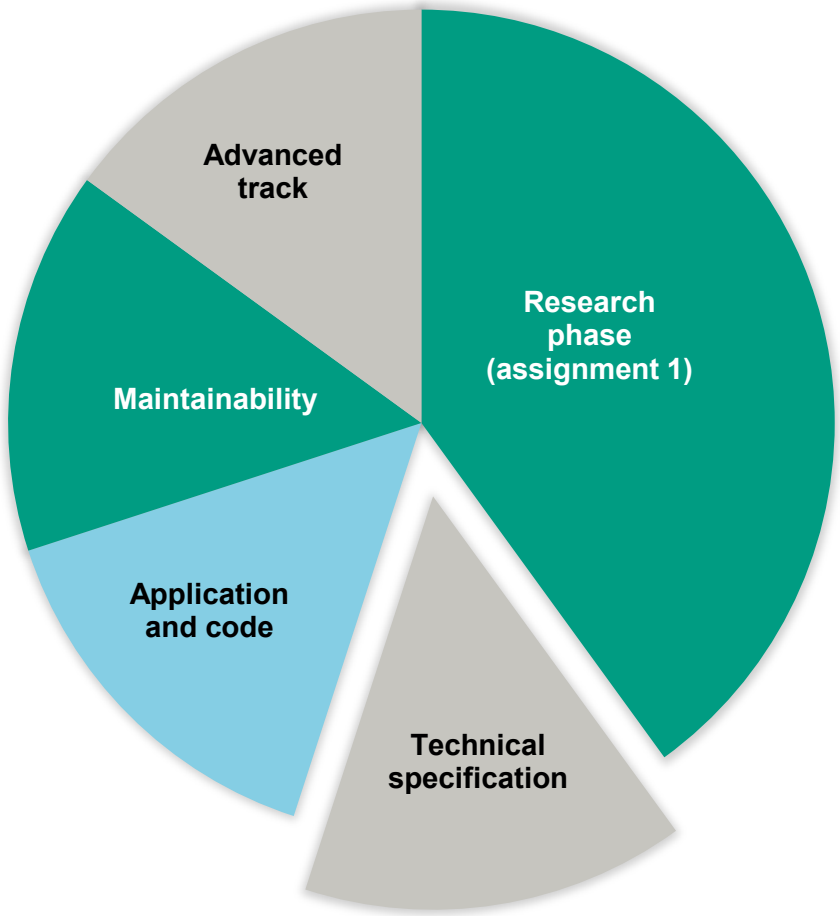
Assignment 2 – rubric

- Technical specification 15
- Application and code 15
- Maintainability 15
- Advanced track 15



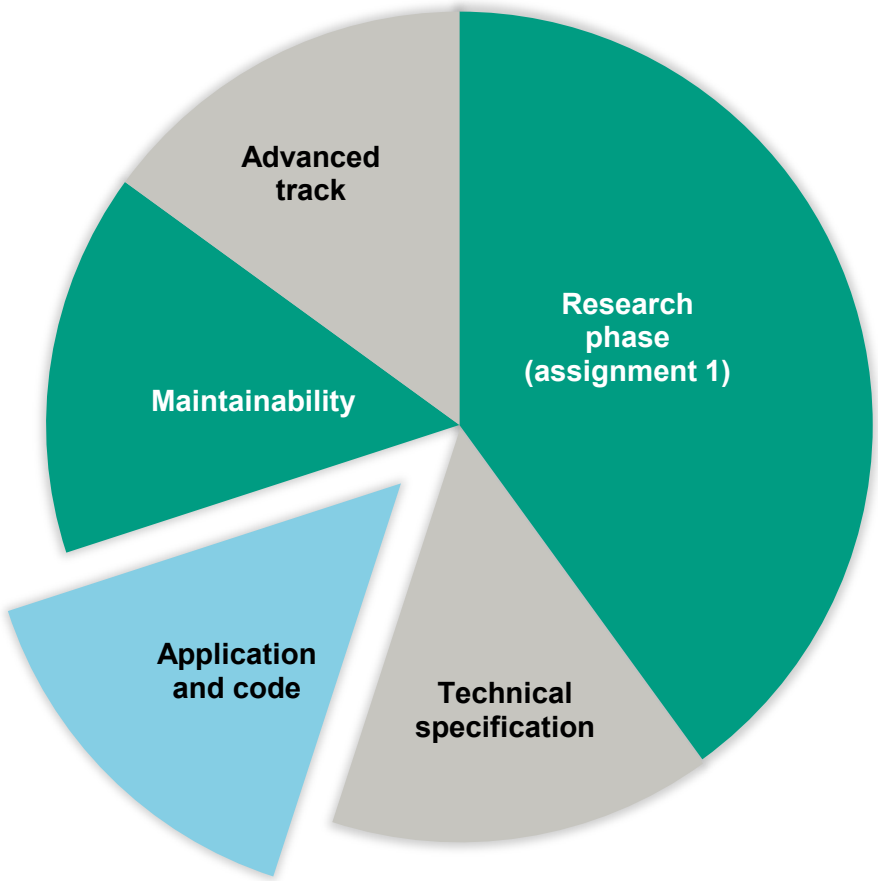
Technical specification

- Why this application (context / description)
- Requirements
- Technical overview:
  - Components
  - How they connect to each other
  - Details where needed
  - Diagrams where needed
  - Choices made, alternatives considered
  - Repeat for subcomponents where needed
- API specification (generated as html is allowed)



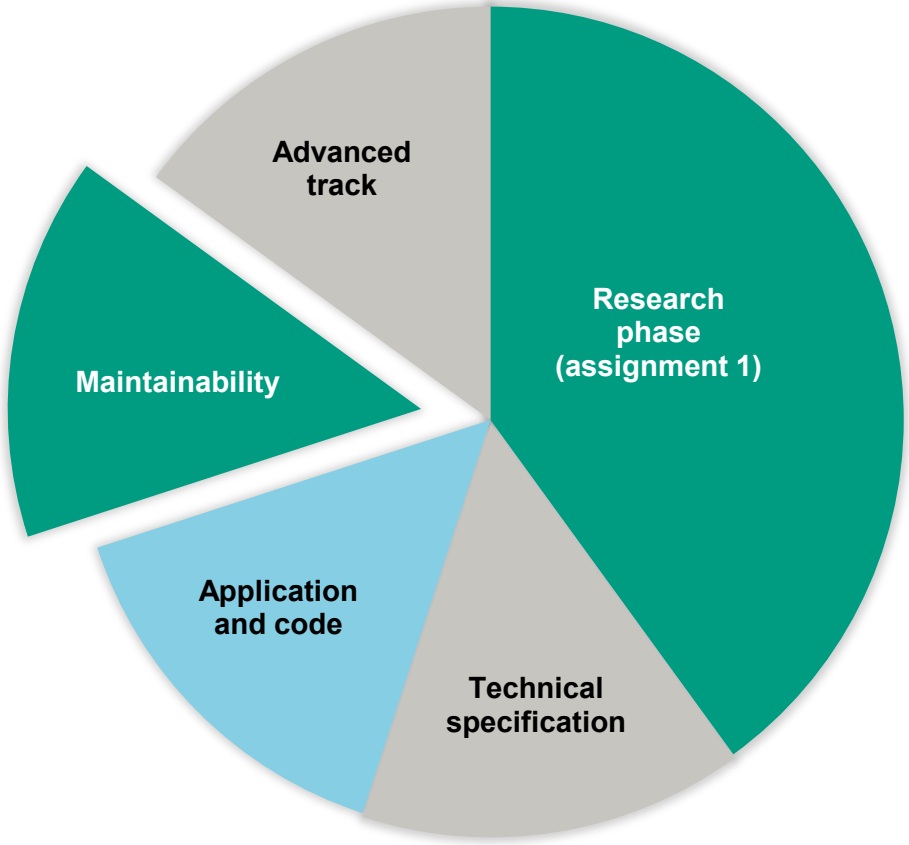
Application and code

- Complete application
- Stable application
- Errorhandling is correct (user errors and technical errors)
- Code quality
  - Neat
  - Structured
  - SoC
  - Readable
  - Navigatable
  - Useful comments



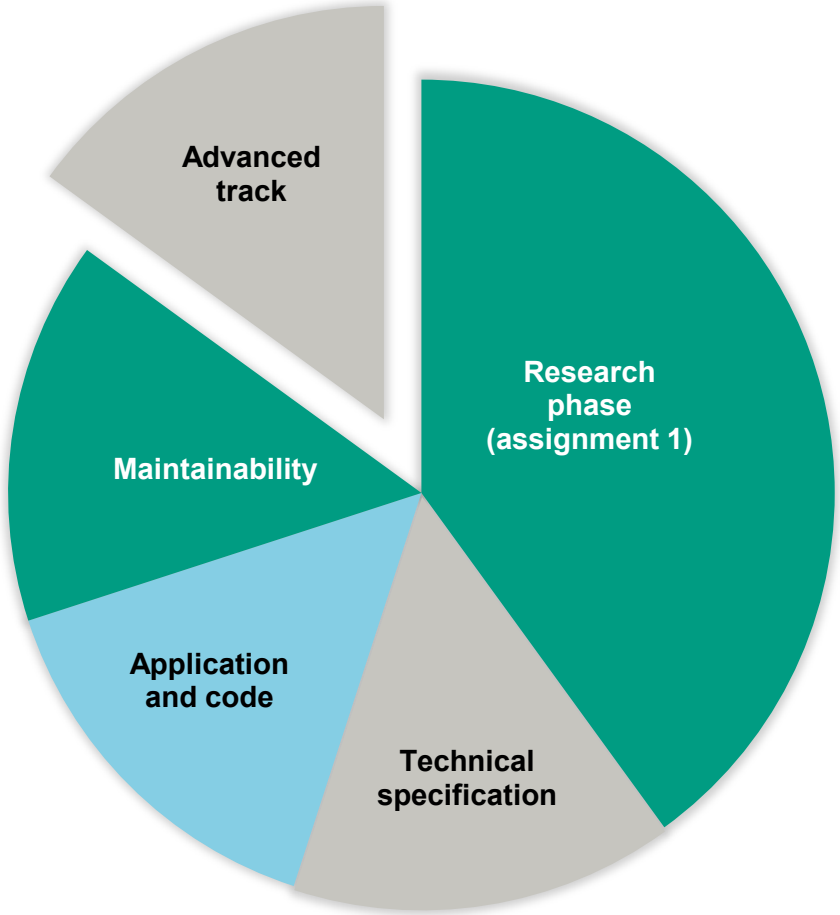
Maintainability

- A manual (a README file) how to start, develop and test the application
- Fully dockerized: `docker-compose up` starts the application without extra steps. (including database and other dependencies)
- Tests
- Testreports



Advanced track

- Depends…
- Documentation:
  - What did you do for the advanced track
  - What does it look like (= proof)
  - How to configure it



Observability

Why observe your production application?

- Users will provide strange and mysterious input
- Clients will report problems
- Application flows will crash
- External services will go down
- Networks will go down
- The application will be too slow
- Servers will be overloaded
- Database queries will take too long
- Rare bugs will occur
- …

What is observability?

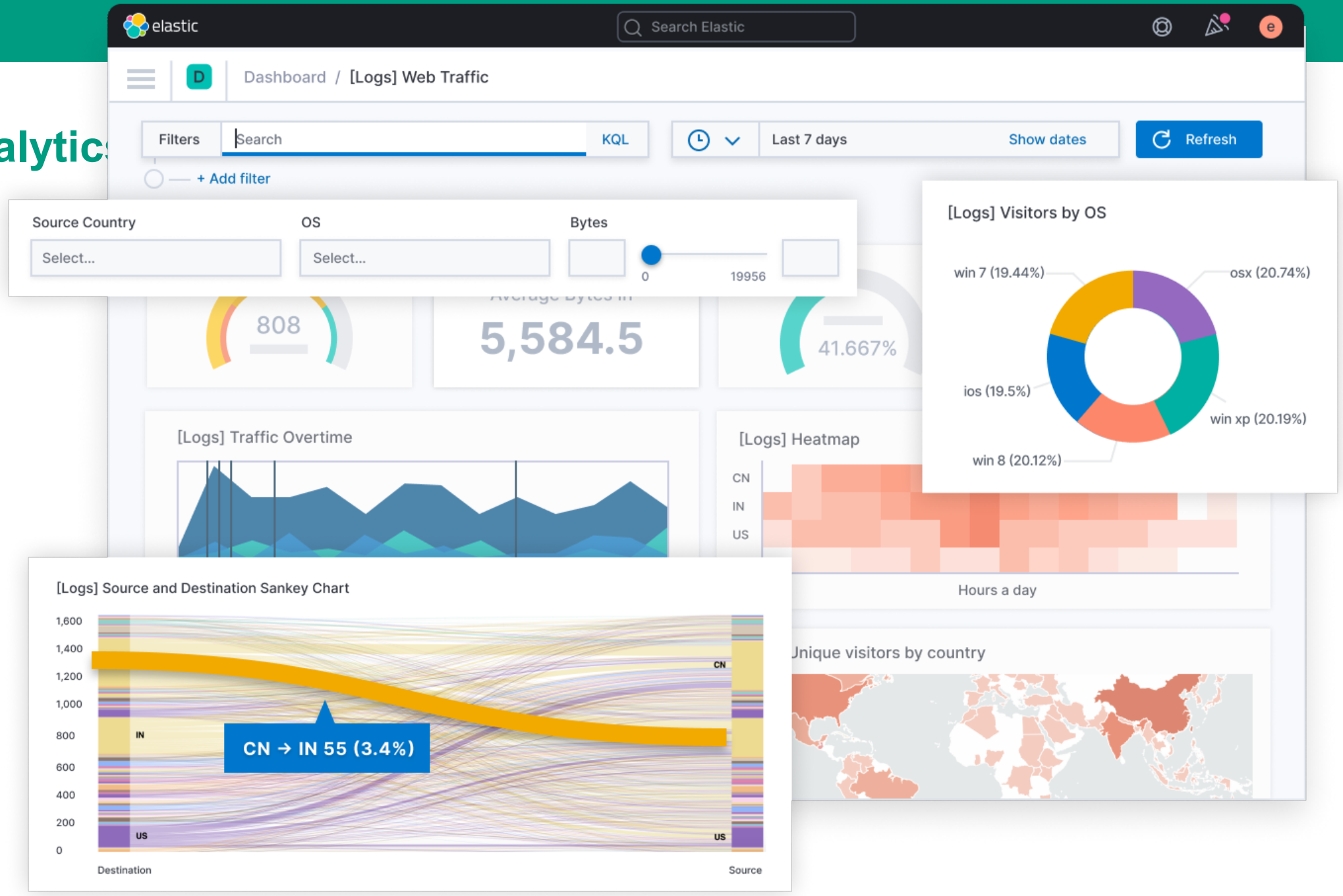
Providing information about the status of the application to an outside observer

- ~~Adding `alert` and `console.log` statements~~
- Structural logging
- Logging analytics
- Performance metrics
- Health checks
- Alerting

Structural logging

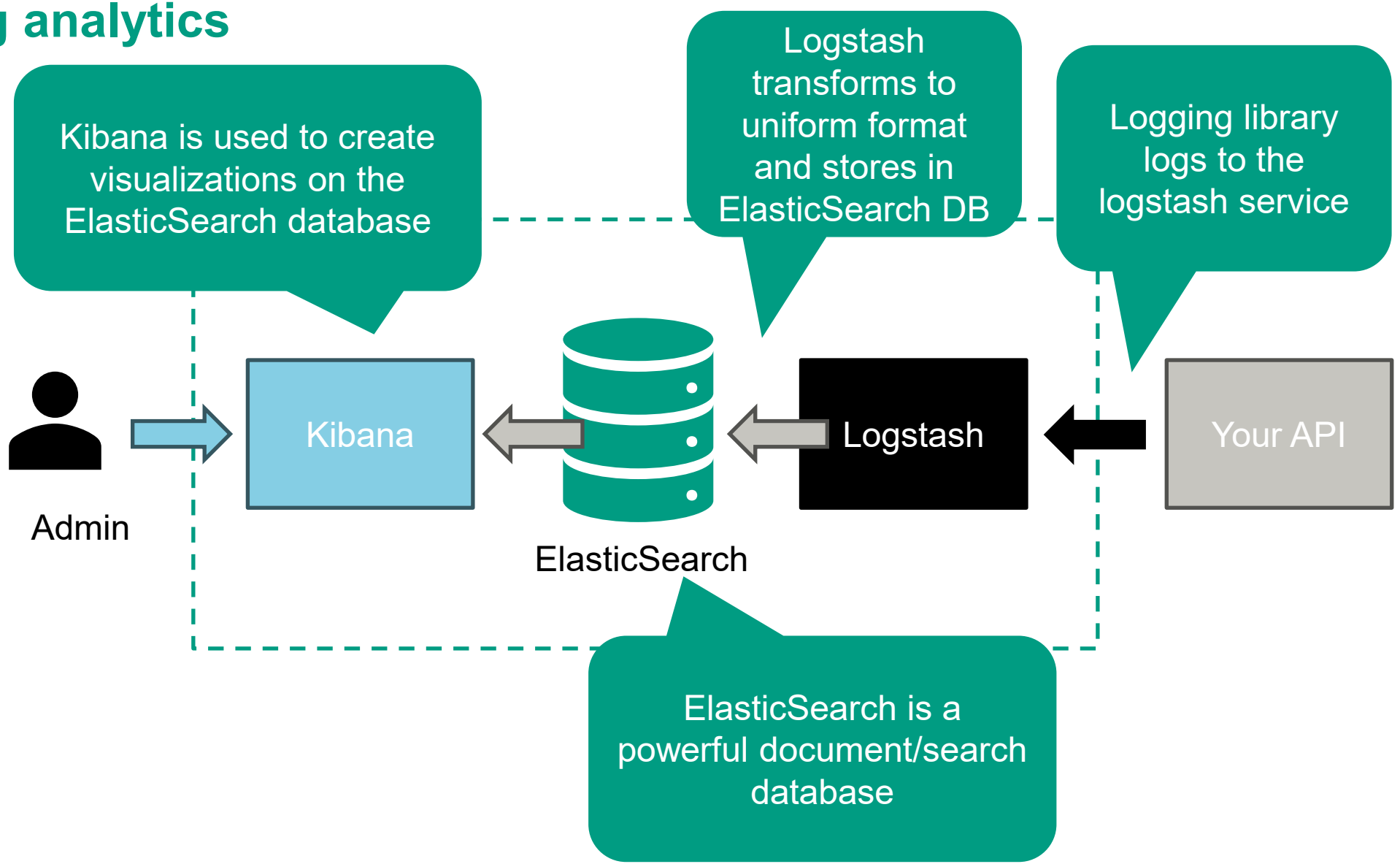
- Add log statements to your code
  - Errors
  - When services go down
  - For additional information
  - …
- Logging to a file
- Add properties: timestamp, application name, module name, user, …
- Use levels: fatal, error, warn, info, debug
- Log to a database for easier analytics

Logging analytics



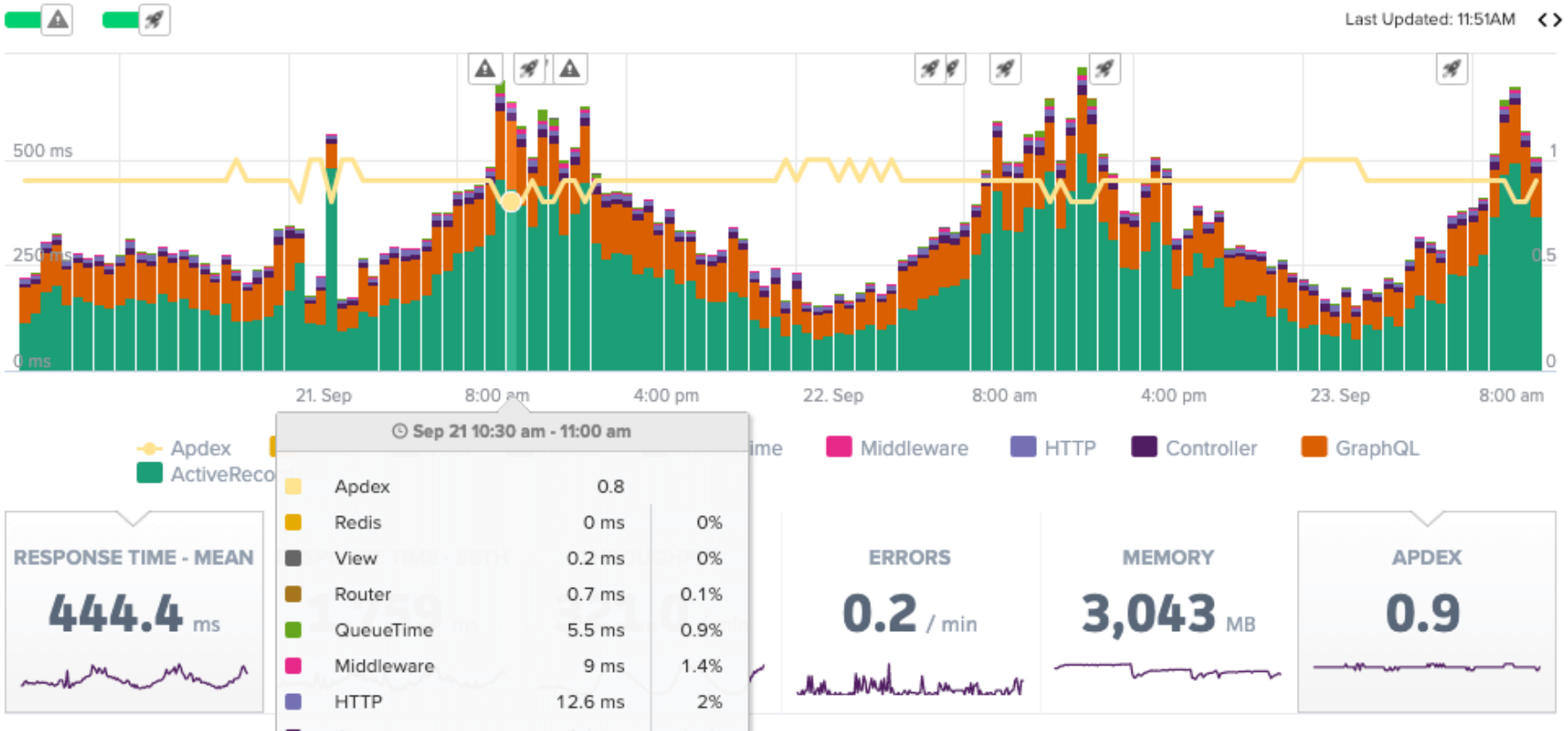
SAXION UNIVERSITY OF APPLIED SCIENCES

Logging analytics



Performance metrics & health checks

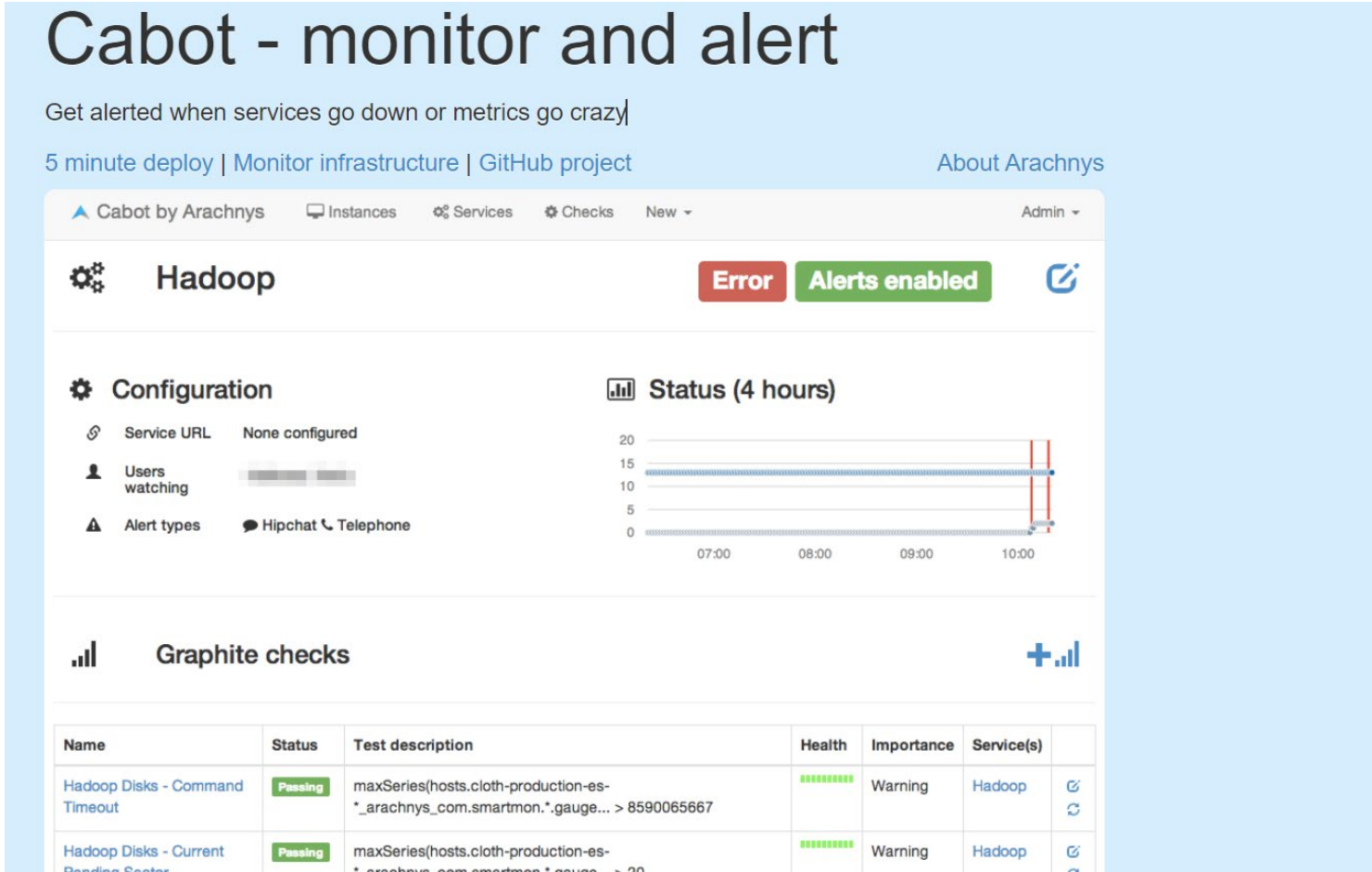
- Error rates of requests
- How long does a request take
- How long does a db query take
- Uptime
- Ping response times
- Resource usage (memory/cpu/disk)



SAXION UNIVERSITY OF APPLIED SCIENCES

Alerting

- Send alerts to people when things go crazy
- Example: cabot



SAXION UNIVERSITY OF APPLIED SCIENCES

Do it yourself

- Find which logging library your framework has
- Or find a logging library to complement your framework

- Add logging to your application (at least for errors)

Note: projects

- 15-20 projects will be published on blackboard in week 7
- Register for the project phase and choose a project or a team
- In week 9-10 you can contact the client already and start the project
- In week 2.1: go.
- In week 2.9-10: assessment

Now what

This week

- Continue with Assignment 2