Microservice architectures quiz

28.3.2019

Agenda

- **Form 2-3 person groups**

- **I will present some problems**

  - Choose A / B / C

  - Fill in blanks

  - etc.

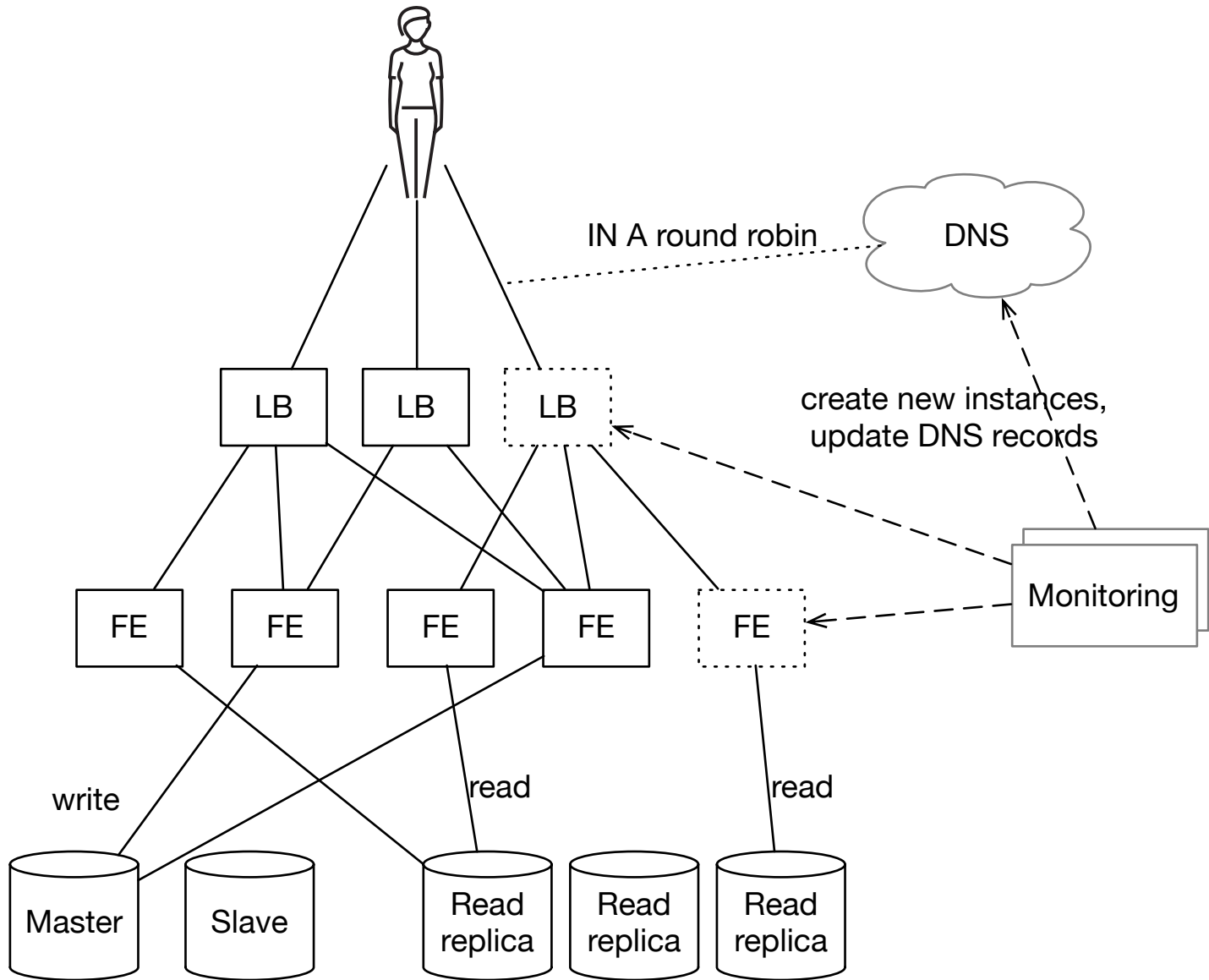- **You'll need to provide rationale**

- **General discussion**
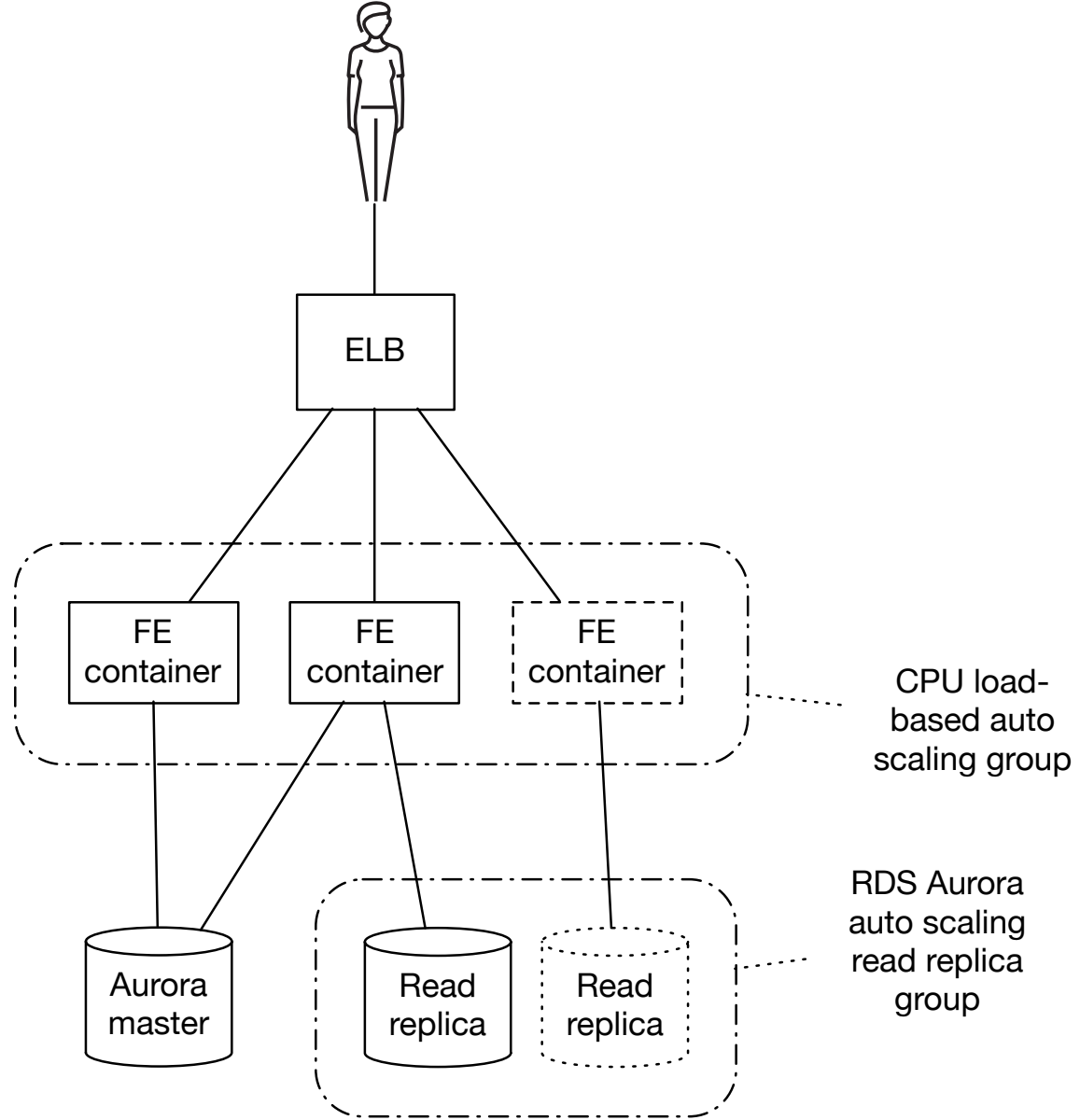
Problem 1: Scalable service

What is the best choice for scalable web service architecture?

- First let's review contestants …

- Discuss your choice in the group for 5 minutes

- Choose A, B or C

- Quick round of "summarize why"

- Deeper discussion

A

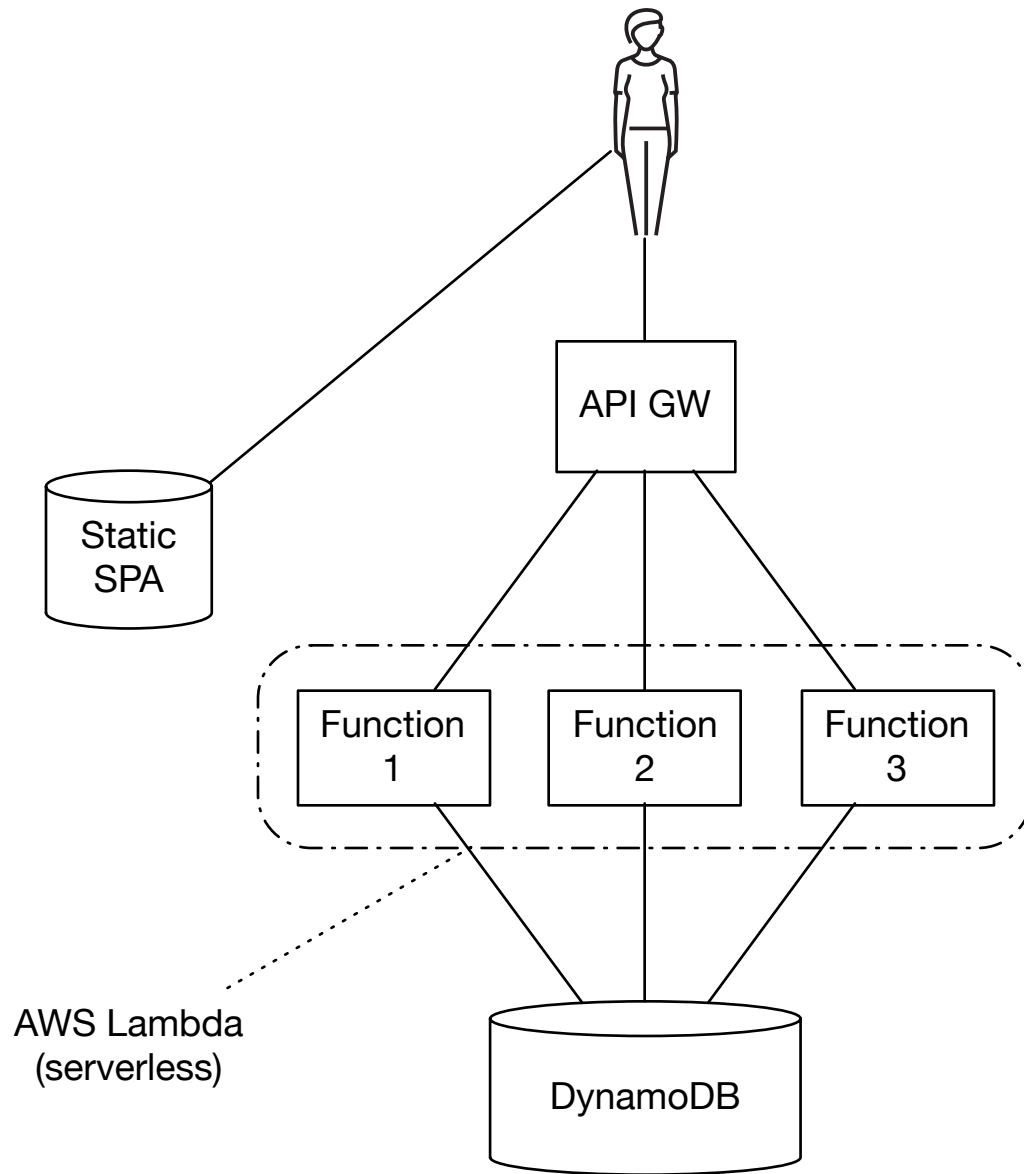IN A round robin .... DNS

LB   LB   LB

create new instances,
update DNS records

FE   FE   FE   FE   FE   Monitoring

write   read   read

Master   Slave   Read replica   Read replica   Read replica

B

ELB

FE container

FE container

FE container

CPU load-based auto scaling group

Aurora master

Read replica

Read replica

RDS Aurora auto scaling read replica group

C

API GW

Static
SPA

Function
1

Function
2

Function
3

AWS Lambda
(serverless)

DynamoDB

A

- IN A round robin ···· DNS
- create new instances, update DNS records
- LB, LB, LB
- FE, FE, FE, FE, FE
- Monitoring
- write
- read
- read
- Master, Slave, Read replica, Read replica, Read replica

B

- ELB
- FE container, FE container, FE container
- CPU load-based auto scaling group
- Aurora master, Read replica, Read replica
- RDS Aurora auto scaling read replica group
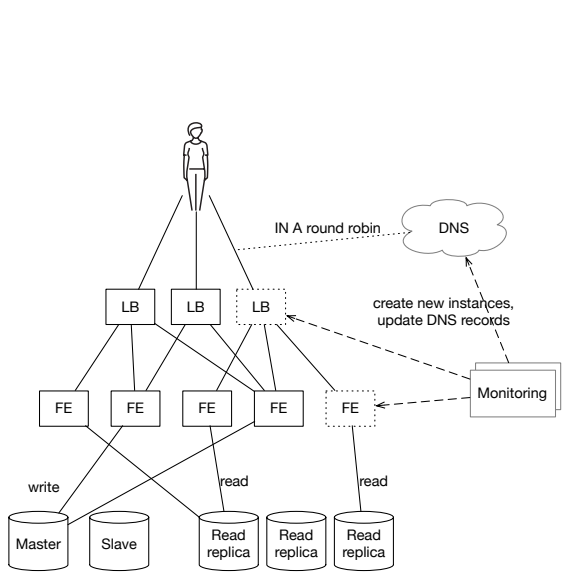
C

- Static SPA
- API GW
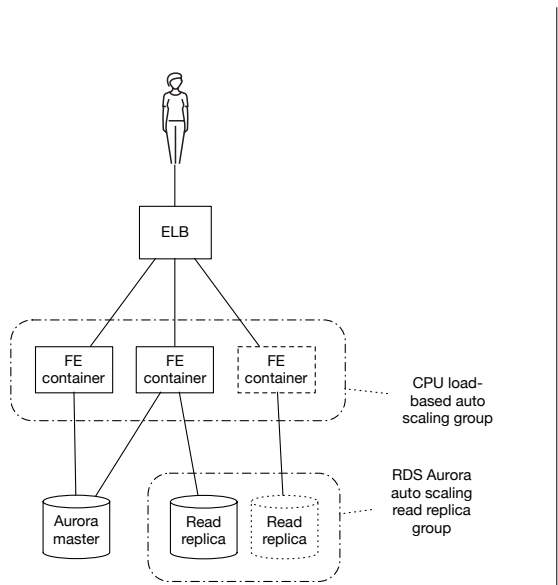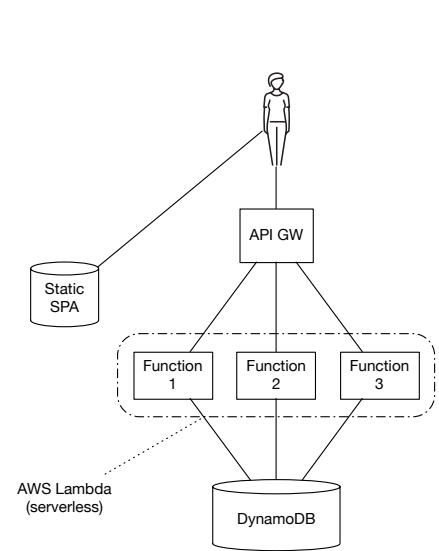- Function 1, Function 2, Function 3
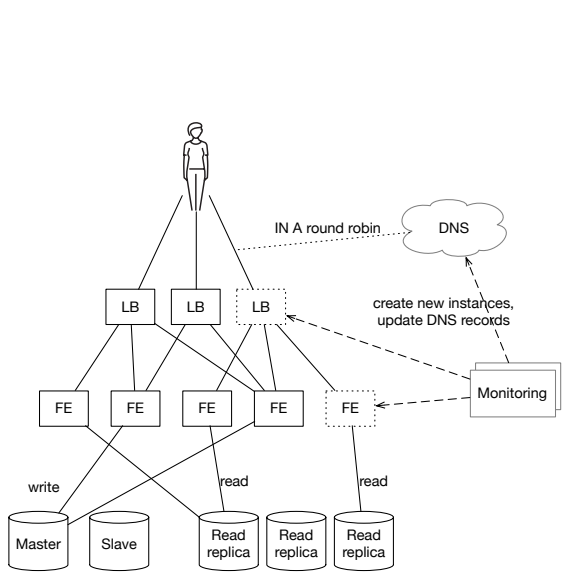- AWS Lambda (serverless)
- DynamoDB
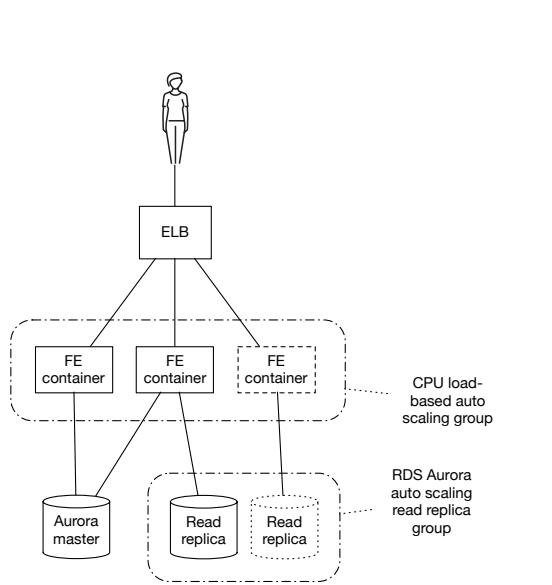
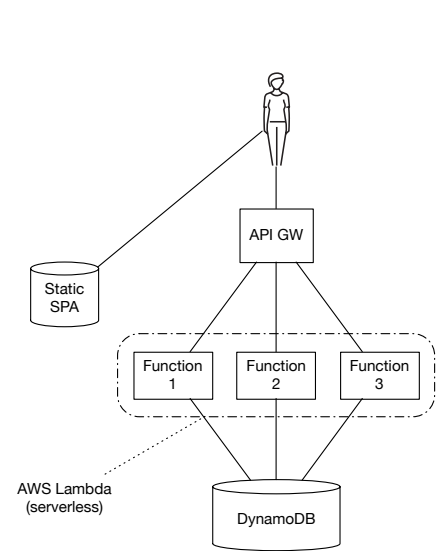# Problem 2: Why one over another?

Why is A better than B and when?

- **Same examples as before**
- **Question is now**
  - What kind of context would preclude your choice?
  - E.g. if your choice was B, then think **when and why** A or C is a **better** choice
- **Discuss in group for 10 minutes**
- **Question time**

A

IN A round robin ···· DNS

create new instances,
update DNS records

LB   LB   LB

Monitoring

FE   FE   FE   FE   FE

write   read   read

Master   Slave   Read replica   Read replica   Read replica

B

ELB

FE container   FE container   FE container

CPU load-
based auto
scaling group

Aurora master   Read replica   Read replica

RDS Aurora
auto scaling
read replica
group

C

Static SPA

API GW

Function 1   Function 2   Function 3

AWS Lambda
(serverless)

DynamoDB

Diagram 1 (Left)

IN A round robin → DNS

DNS ⟵ create new instances, update DNS records ⟵ Monitoring

LB — LB — LB

Monitoring

FE — FE — FE — FE — FE

write | read | read

Master | Slave | Read replica | Read replica | Read replica

Diagram 2 (Middle)

ELB

FE container | FE container | FE container → CPU load-based auto scaling group

Aurora master | Read replica | Read replica → RDS Aurora auto scaling read replica group

Diagram 3 (Right)

API GW

Static SPA

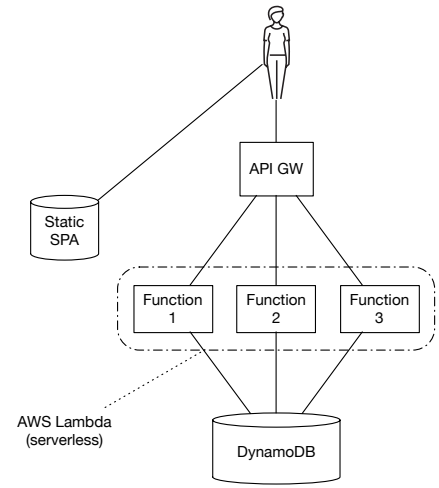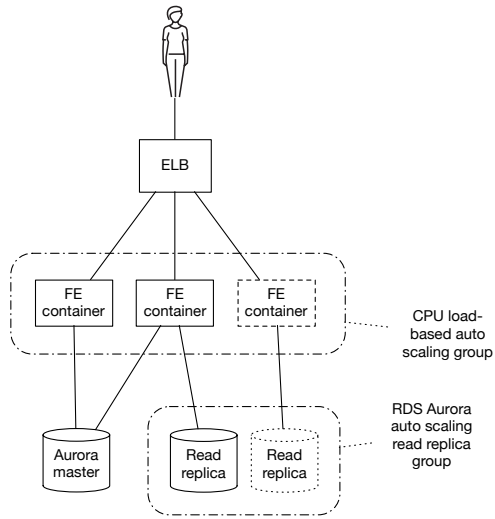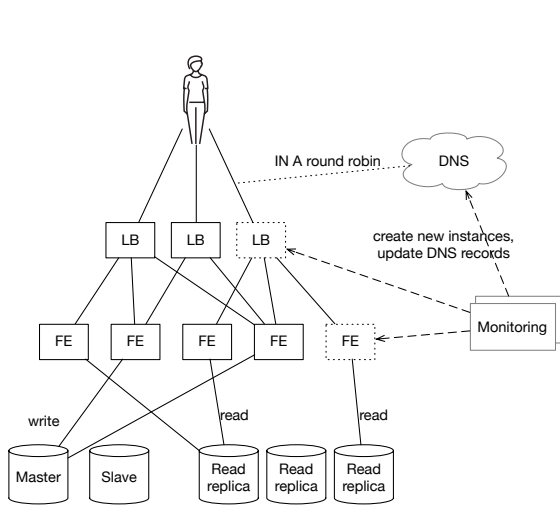Function 1 | Function 2 | Function 3

AWS Lambda (serverless)

DynamoDB

Good when?

Bad when?

Problem 3: Large data

How to handle large amounts of incoming data?

- **Method: design your own**

  - Each group gets a slice of whiteboard

  - Design and draw an architecture for solving the problem

  - 10 minutes later …

  - Each group gets to describe their design and describe their rationale and assumptions they used

- **So …**

  - Let's **first** look at the problem statement (need projector!)

  - **Then** work on whiteboard

How to handle large amounts of incoming data?

- **Problem statement**

  - System ingesting data from customers at a rate of several GB/hour

  - Need to process incoming data for daily summaries

  - Need to understand data ingestion per customer (billing)

- **Note that the following are <u>not</u> specified (your call)**

  - What kind of data and in how large batches

  - How many customers and what customer churn

  - Length of data retention

  - Etc.

Problem 4:
What does this do?

What does this do?

- **Box-and-lines diagram**

- **No labels in boxes**

- **Your mission, if you choose to accept it,**

  (but you don't really have a choice due to the narrative imperative)

- **is to think about what this system could do and fill in descriptive labels into the boxes**

- **Discuss in group 10 minutes**

- **Quick description of what you think the system is doing**

- **Then another round with more depth**

LOONEY TUNES

"That's all Folks!"

almost

Next week

- **Final lecture**
- **Summarizing course contents**
- **Maybe some hints on exam**
- **Question time!**
  - Any questions related to course are welcome
  - Please prepare in advance (need to know how many at some point): e.g. write down
  - Paper, email, slack, … (we'll get more during the lecture)

- **Exam 11.4. (TU7/TUAS) 16:30-19:30**
- **Personal coursework deadline 14.4. 23:59**
  - Share your repository in version.aalto.fi to me or course group
  - Submit repository URL and commit hash via MyCourses (https://mycourses.aalto.fi/mod/assign/view.php?id=409255)