



© D. Bermbach

# Fog Computing

Bermbach | Recap

---

Breakout sessions

# RECAP: FROM CLOUD TO FOG

# Breakout Sessions

Discuss in small groups the following questions:

*Wanna hang out sometime soon? (seriously, get to know each other)*

*Name one use-case each: What is a good example for...*

- *A Cloud-only application?*
- *An Edge-only application?*
- *A Fog application?*

15mins for discussion and choosing **one** speaker

Afterwards, the speakers each present their result/answer and briefly discuss.

# Recap

Discuss in small groups the following questions:

*Is Fog Computing essentially a Cloud 2.0 and will gradually replace the cloud?*

# Recap

## *Group 1 + 2:*

Revisit the usecases presented in the last lecture. Can you think of other usecases which would benefit from fog computing?

## *Group 3 + 4:*

Which of the challenges for fog adoption are the dominant ones in your opinion? Can you think of other challenges?

Breakout sessions

# RECAP: DATA DISTRIBUTION

# Recap

Group 1 + 2:

What are characteristics of an IoT application use case in which *Selecting RPs close to the Publishers* is more efficient than *Selecting RPs close to the Subscribers*.

Group 3 + 4:

What problems can arise when brokers have very heterogenous resources if inter-broker message routing is based on the *broadcast groups approach*?  
How could the approach be changed to mitigate such problems?

# Recap

Group 1:

What are typical application use cases which will benefit from pub/sub?

Group 2:

When (and why) would you use synchronous or asynchronous communication? Think about edge-to-edge, edge-to-cloud, cloud-to-cloud, etc.

Group 3 + 4:

What problems can arise when brokers have very heterogeneous resources and inter-broker message routing is based on the *broadcast groups approach*? How could the approach be changed to mitigate such problems?



Breakout sessions

# RECAP: DATA MANAGEMENT

# Recap

*Group 1 + 2:*

Think of one application use case each in which you would configure DisGB to place RPs either close to publishers or close to subscribers.

*Group 3 + 4:*

Does replication always increase availability?

# Recap

## Group 1:

Which consistency model group (data-centric or client-centric) would you as an application developer prefer for a Service Level Agreement? Why?

## Group 2-4:

Think about a concrete fog application use case (e.g., lighting in a smart home) and describe how a violation of *Monotonic Reads* consistency (group 2), *Read Your Writes* consistency (group 3), *Monotonic Writes* consistency (group 4) would materialize for clients/cause problems. If you have time left, think also about *Write Follows Read* consistency. Feel free to create a quick draft for a slide (don't spend a lot of time on this).

# Recap

Discuss in your groups the four different replication techniques and think about storage needs of different kinds of data in fog use cases. Which replication technique would you use for which kind of data and where (not: how) would you place replicas (e.g., edge-only, cloud-only)?

# Recap

Thinking about “mixed” replication techniques (e.g., sync update everywhere on a set of edge nodes plus async replication to the cloud = async primary copy in which the collection of edge nodes acts as the primary), which of the replica placement strategies that we talked about might prove useful?

Breakout sessions

# RECAP: PLATFORMS & APPLICATIONS

# Recap discussion round

Every group: Pick one of the presented systems (Lean OpenWhisk, tinyFaaS, NanoLambda) and describe an example application use case in which you would use the respective system but in which you could not use any of the other two systems.

Breakout sessions

# RECAP: TESTING & BENCHMARKING



# Recap discussion round

Where and how would you deploy a distributed application across cloud/edge/fog-based FaaS nodes to assert fault-tolerance and availability of such an application?

- Which requests are served where?
- What happens when a FaaS node fails?
- What happens when a FaaS node is overloaded?
- ...

# Recap discussion round

Your company is offering a fog application and high availability is of utmost importance.

- Which priority do you respectively see for benchmarking vs. testing vs. monitoring?
- Which of the three approaches do you use for what (and possibly how)?

# Recap discussion round

Revisit the design and implementation objectives for benchmarks. Do you see any problems with the DeFog case study?