

beyond
Jakarta EE
and microprofile

adam-bien.com

"It's not work if you like it"
...so I never worked. #java



adam-bien.com

learn once,
never migrate

adambien.blog
airhacks.io
airhacks.tv
airhacks.news
airhacks.fm

adam-bien.com

press.adam-bien.com

Real World Java EE Night Hacks

Dissecting the Business Tier

[Iteration One]



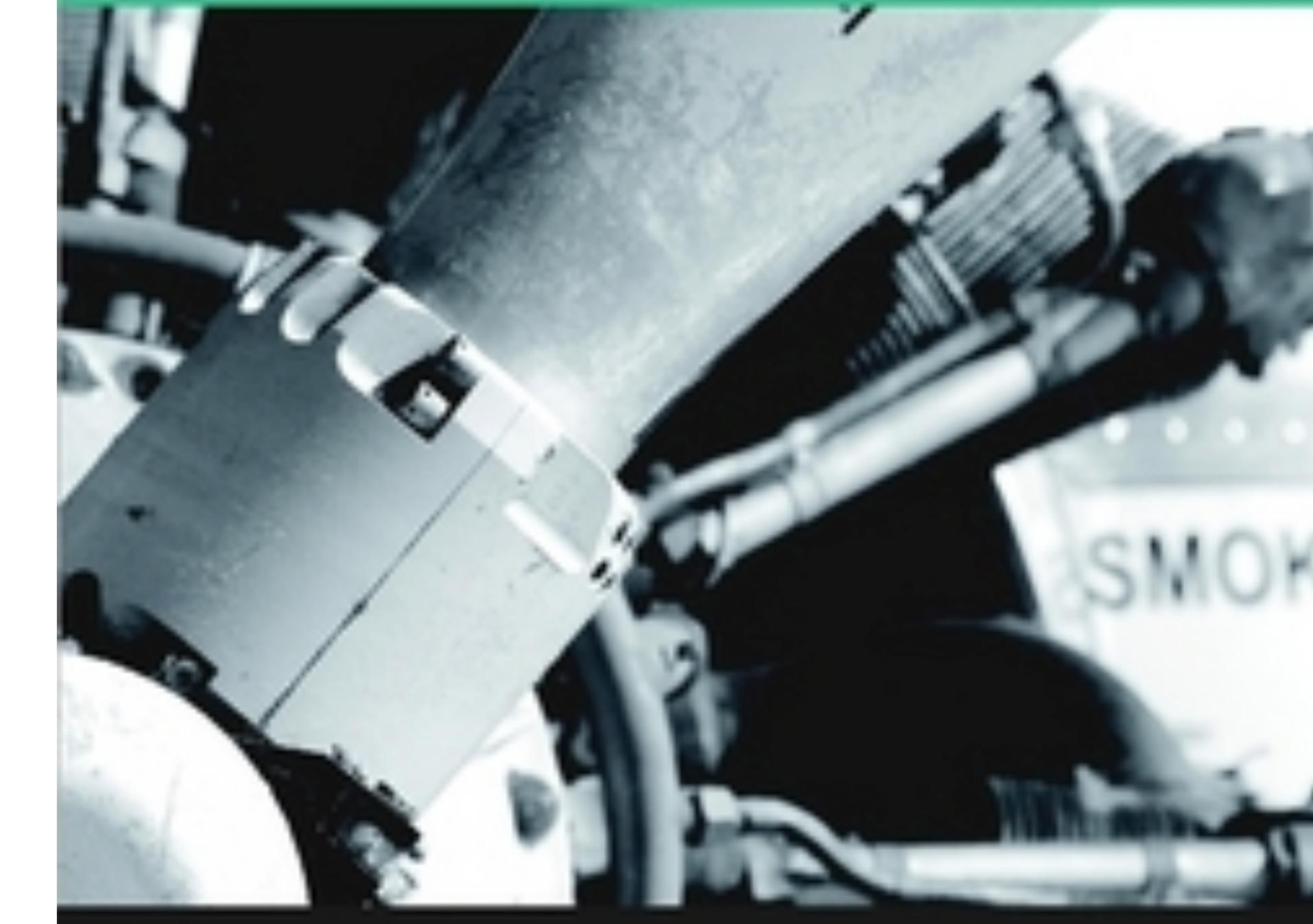
Adam Bien

Foreword by James Gosling

REAL WORLD

JAVA EE PATTERNS

RETHINKING BEST PRACTICES



adam-bien.com

airhacks.TV

Munich (MUC) Airport Web Workshops for Java Developers, Summer 2020:

WebComponents Kickstarter, June 30th, 2020

partially also available as: [Streaming / Download Edition]

Building Apps with WebComponents, July 1st, 2020

available as: [Streaming / Download Edition]

Munich (MUC) Airport Workshops, Winter 2019:

Jakarta EE / MicroProfile Microservices, December 10th, 2019

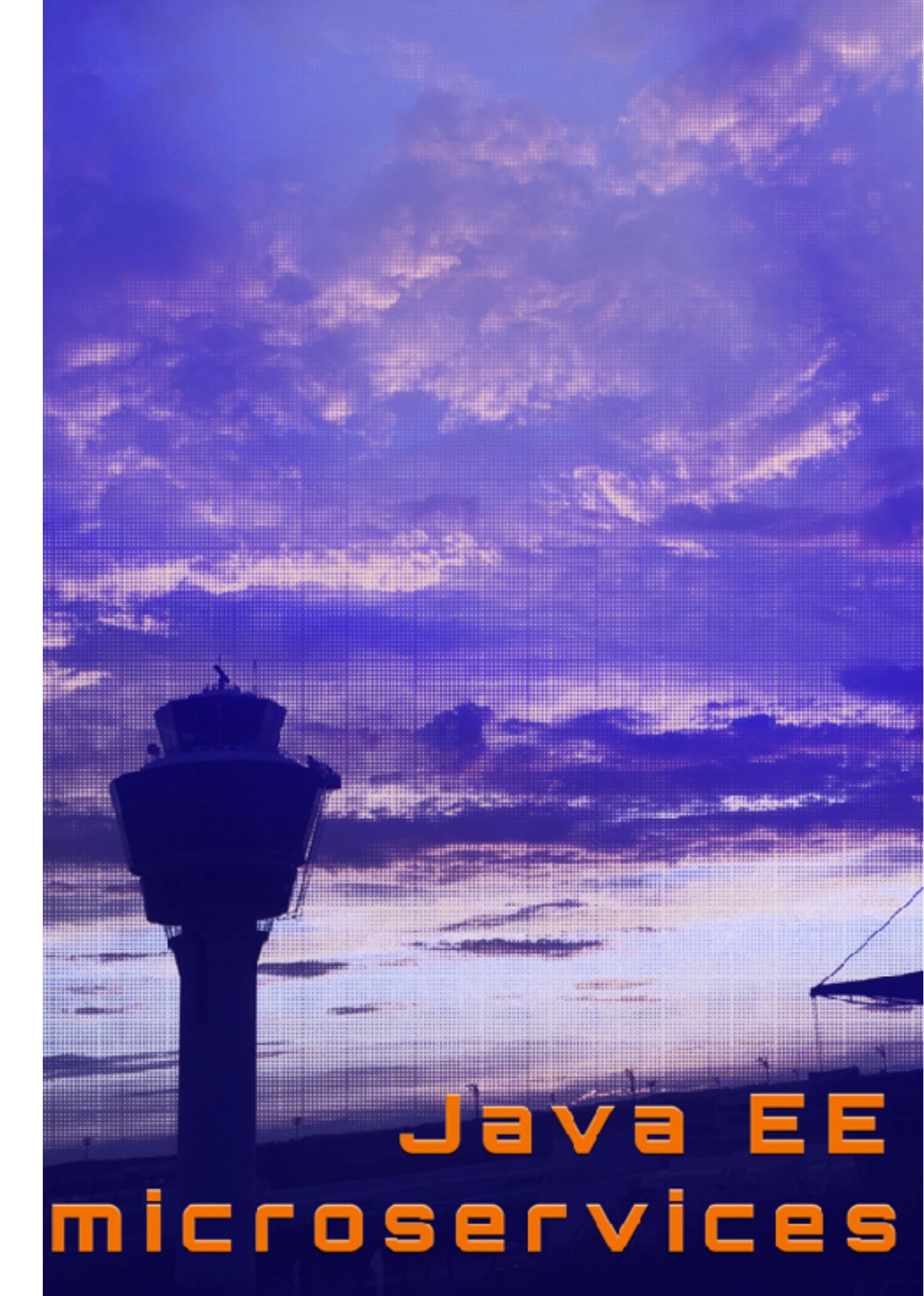
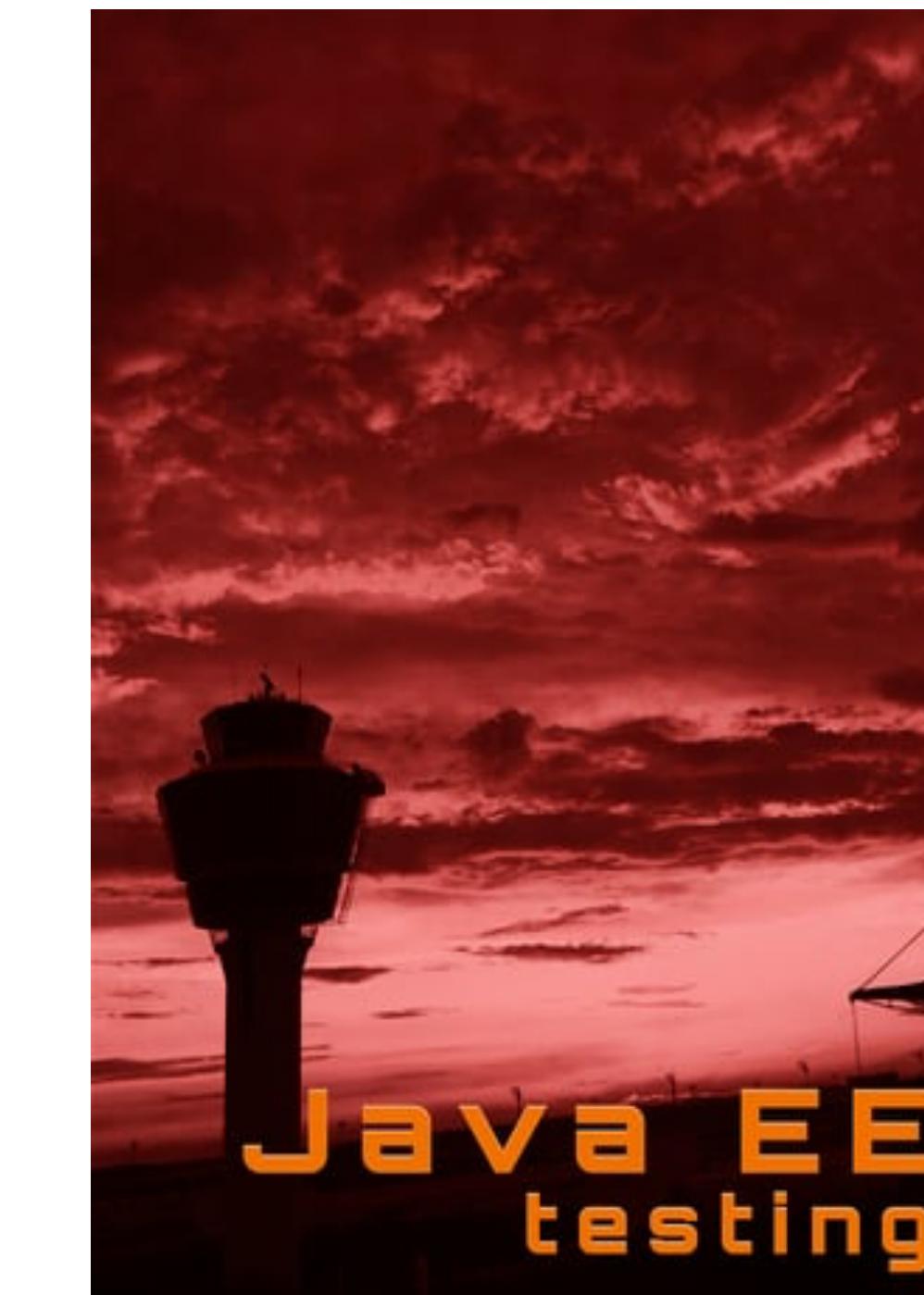
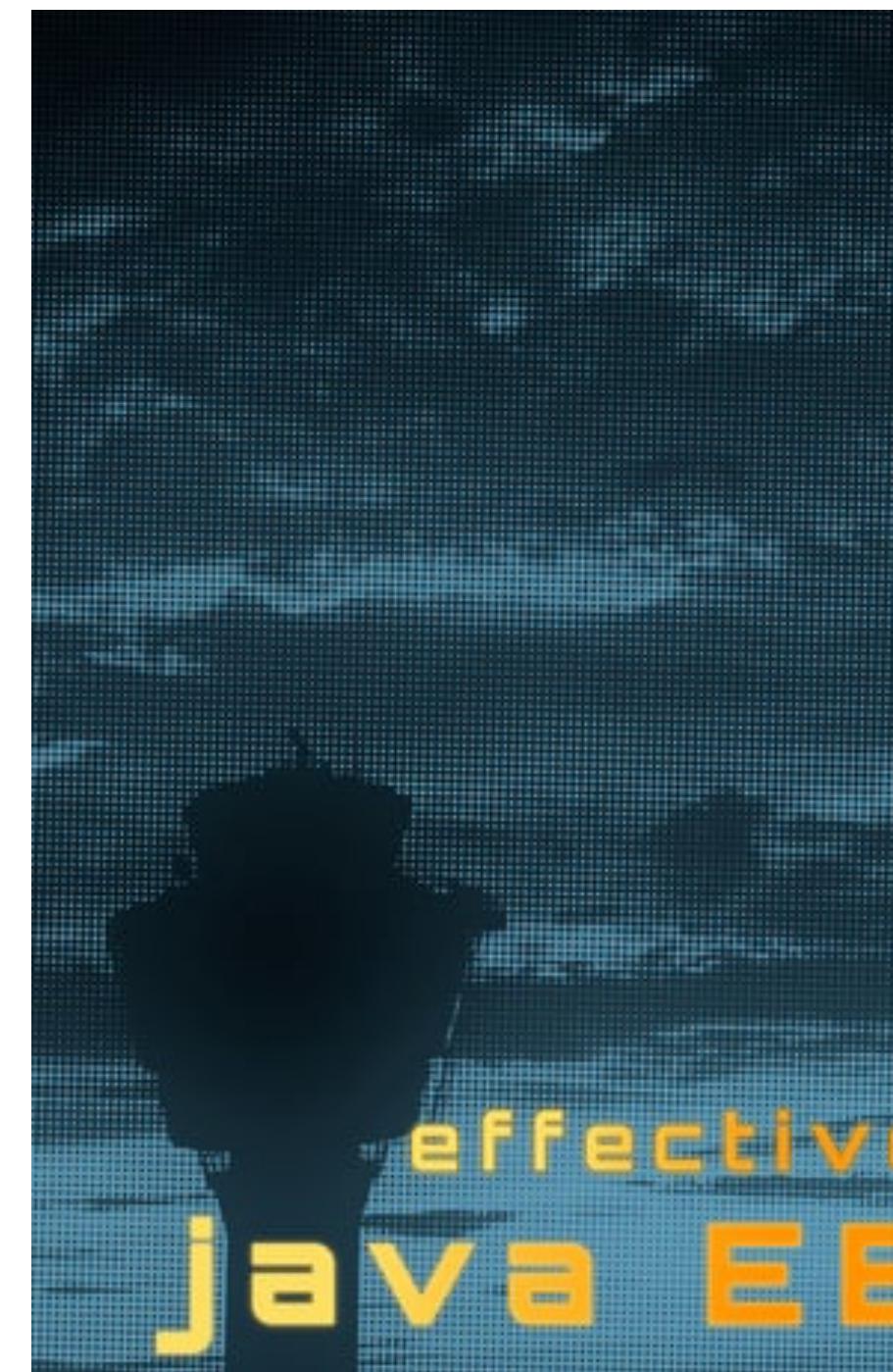
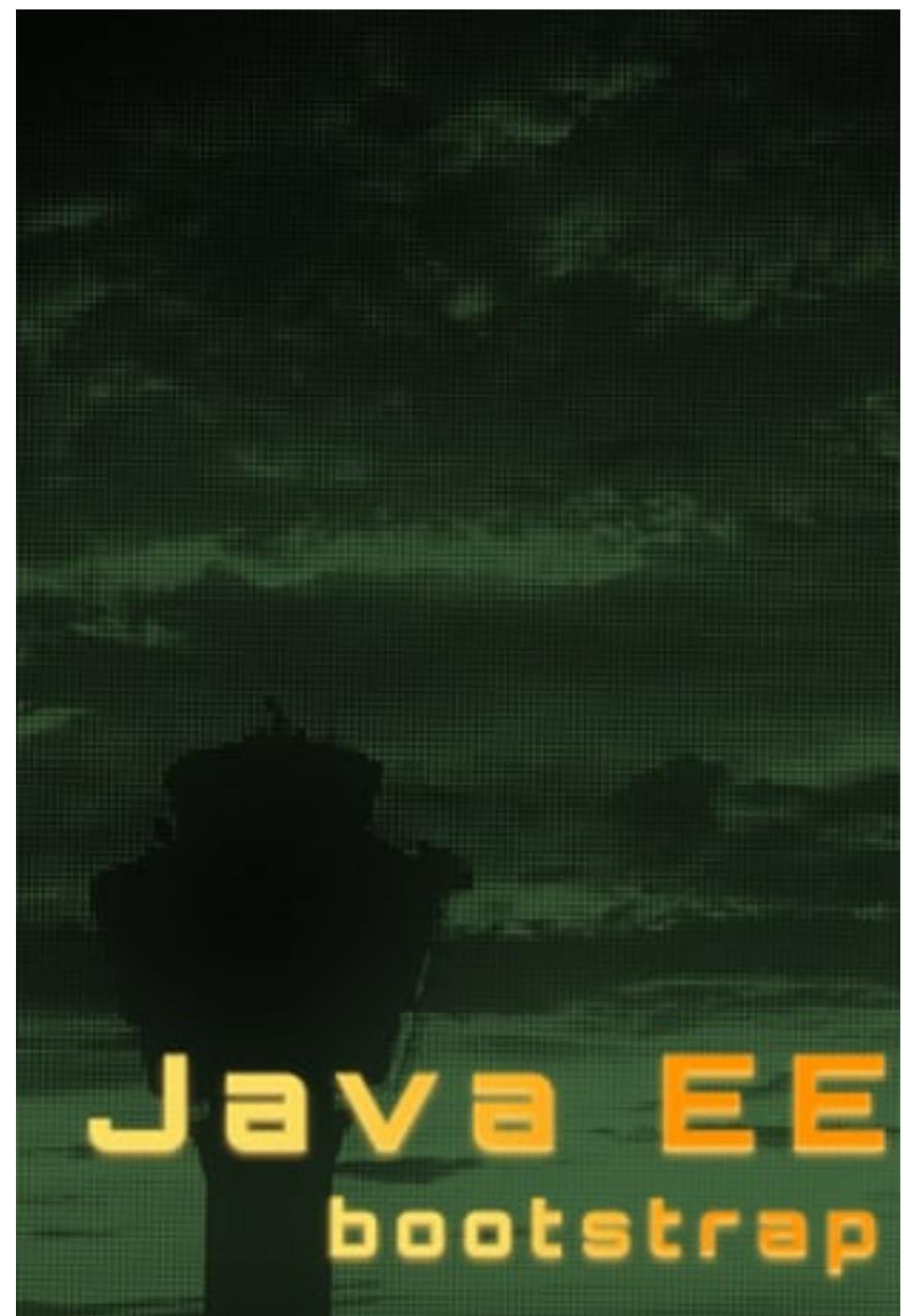
also available as: [Streaming / Download Edition]

Jakarta EE, Microprofile and Clouds, December 11th, 2019

Beyond Boring Jakarta EE and JDK 12--From NoSQL, over
Reactive to Fibers, December 12th, 2019

NEW:

Streaming Architectures, December 13th, 2019



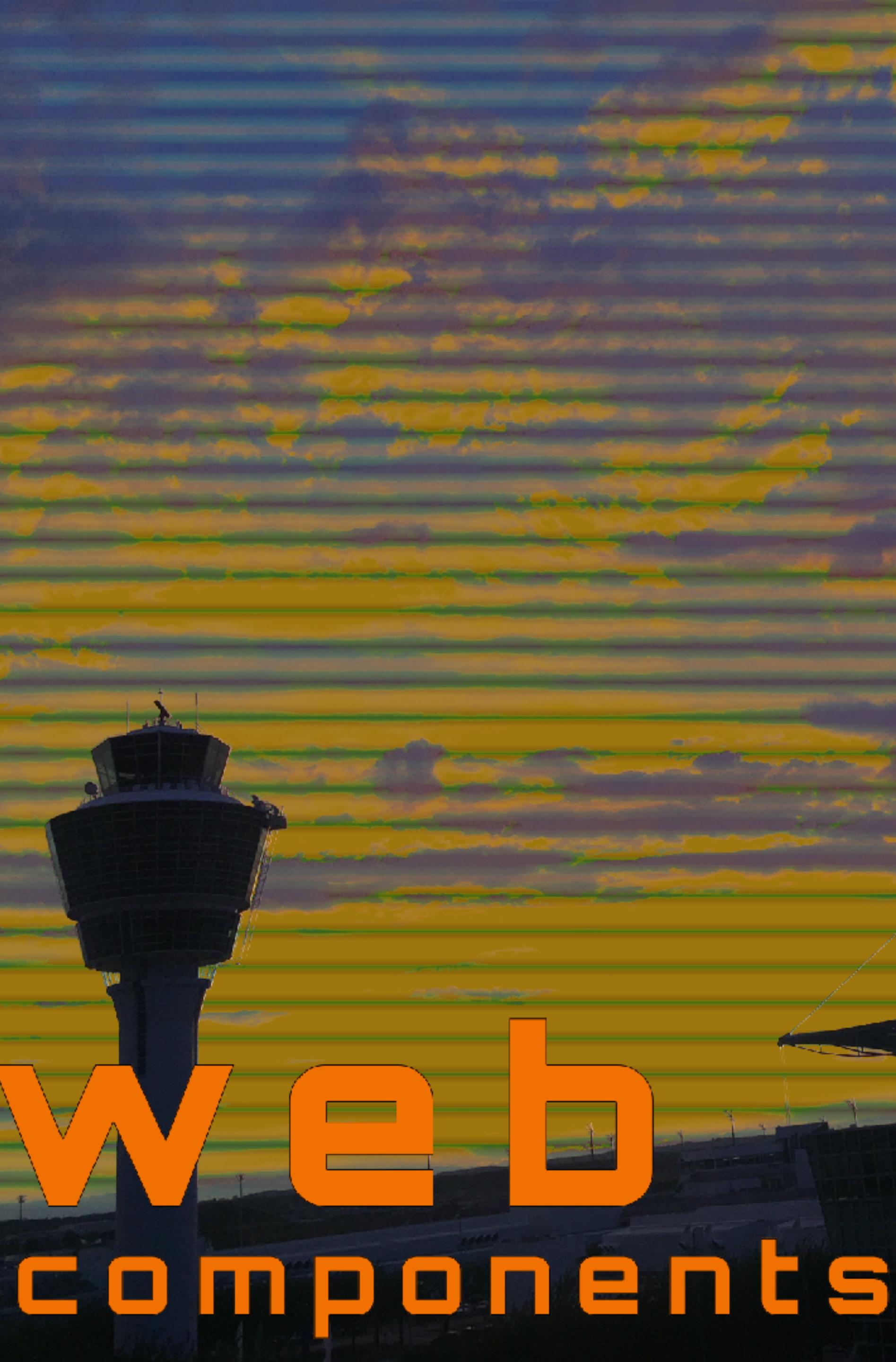
airhacks.io

<http://webstandards.training>

A photograph of a night sky filled with horizontal clouds. In the foreground, the dark silhouette of a lighthouse tower stands prominently against the lighter clouds.

web
{standards}
igniter

<http://webcomponents.training>



Munich (MUC) Airport Web Workshops for Java Developers, Summer 2020:

WebComponents Kickstarter, June 30th, 2020

partially also available as: [Streaming / Download Edition]

Building Apps with WebComponents, July 1st, 2020

available as: [Streaming / Download Edition]

Munich (MUC) Airport Workshops, Winter 2019:

Jakarta EE / MicroProfile Microservices, December 10th, 2019

also available as: [Streaming / Download Edition]

Jakarta EE, Microprofile and Clouds, December 11th, 2019

Beyond Boring Jakarta EE and JDK 12--From NoSQL, over
Reactive to Fibers, December 12th, 2019

NEW:

Streaming Architectures, December 13th, 2019

AI

RBAC

Runtimes

Logging

Read phenomena

[https://en.wikipedia.org/wiki/Isolation_\(database_systems\)#Read_phenomena](https://en.wikipedia.org/wiki/Isolation_(database_systems)#Read_phenomena)

Dirty Reads

- A dirty read (aka uncommitted dependency) occurs when a transaction is allowed to read data from a row that has been modified by another running transaction and not yet committed.

Non-repeatable reads

- A *non-repeatable read* occurs, when during the course of a transaction, a row is retrieved twice and the values within the row differ between reads.

Phantom reads

- A *phantom read* occurs when, in the course of a transaction, new rows are added or removed by another transaction to the records being read.

Isolation Levels

- Serializable
- Repeatable reads
- Read committed
- Read uncommitted

Understanding CAP in Java EE context

**CAP vs. FLP (Fischer,
Lynch and Paterson)**

In an asynchronous network where messages may be delayed but not lost, there is no consensus algorithm that is guaranteed to terminate in every execution for all starting conditions, if at least one node may fail-stop.

Paxos, Multi Paxos, Fast Paxos

RAFT

Raft is a **consensus** algorithm designed as an alternative to **Paxos**. It was meant to be more understandable than Paxos by means of separation of logic, but it is also formally proven safe and offers some additional features.^[1] Raft offers a generic way to distribute a **state machine** across a **cluster** of computing systems, ensuring that each node in the cluster agrees upon the same series of state transitions.

[https://en.wikipedia.org/wiki/Raft_\(computer_science\)](https://en.wikipedia.org/wiki/Raft_(computer_science))

**Principles: ACID, BASE, Copy
On Write, Lazy / Eager
Loading, Optimistic /
Pessimistic Locking**

Non-Relational DBs (NoSQL)

NoSQL

- Key Value
- Document
- Graph

Clustering

- Don't Distribute
- Replication
- Distribution (Sharding)
- P2P
- Master-Slave

Write Consistency

- Write-write conflicts
- Conditional updates
- Session consistency
- Replication durability (HTTP Session)
- Locks

Read Consistency

- Read-write conflict
- Replication consistency
- Conditional updates

Transactions

- System Transactions
- Business Transactions
- Simplicity -> System TX == Business TX

**Principles: ACID, BASE, Copy
On Write, Lazy / Eager
Loading, Optimistic /
Pessimistic Locking**

Versioning

- Counter
- Timestamp
- UUID
- Content based: Hash (see Merkle Tree)
- Composite
- History: vector clocks

Map-Reduce

- Inter-Aggregate computations
- 2 Phases:
 - Map: parallel filtering, extraction, transformation
 - Reduction: Map results transformed into single values

Dynamo

Consistent Hashing

Gossip Protocol

Vector Clocks

Hinted Writes

Read Repair

Merkle tree

Serialization,
Externalization and
optimizations

Graal VM

Off-heap memory

**Column and row based
stores**

What is map-reduce?
Samples and use cases.

Distributed and persistent caches

- Infinispan (JBoss Cache)
- Hazelcast

Esoteric DB Integrations

Serverless

Munich (MUC) Airport Web Workshops for Java Developers, Summer 2020:

WebComponents Kickstarter, June 30th, 2020

partially also available as: [Streaming / Download Edition]

Building Apps with WebComponents, July 1st, 2020

available as: [Streaming / Download Edition]

Munich (MUC) Airport Workshops, Winter 2019:

Jakarta EE / MicroProfile Microservices, December 10th, 2019

also available as: [Streaming / Download Edition]

Jakarta EE, Microprofile and Clouds, December 11th, 2019

Beyond Boring Jakarta EE and JDK 12--From NoSQL, over
Reactive to Fibers, December 12th, 2019

NEW:

Streaming Architectures, December 13th, 2019

Thank You!

blog.adam-bien.com
twitter.com/AdamBien