EVENTLETS - Containers for Generic Event-driven Tasks

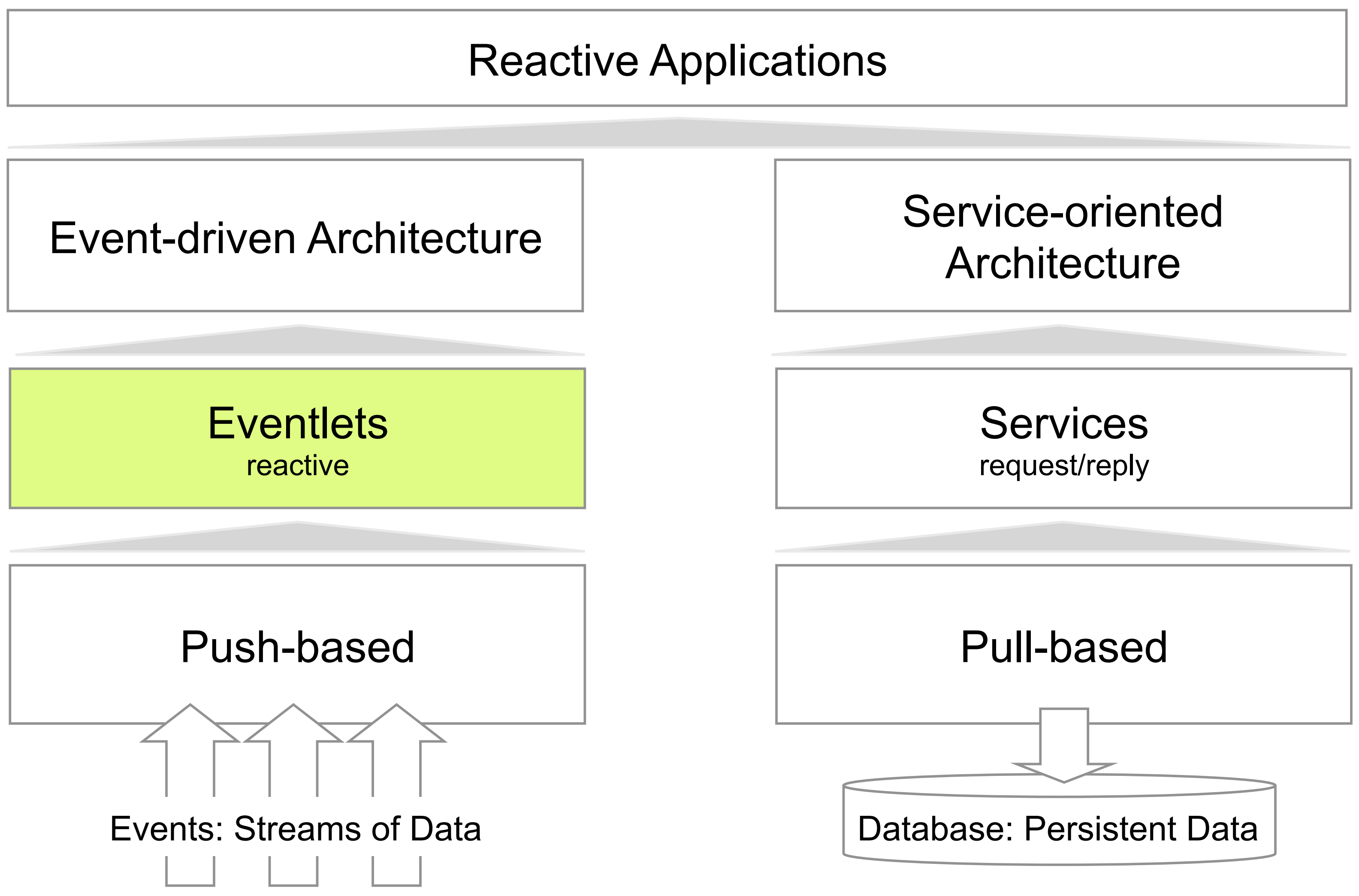
TECHNISCHE UNIVERSITÄT DARMSTADT

DVS

Stefan Appel

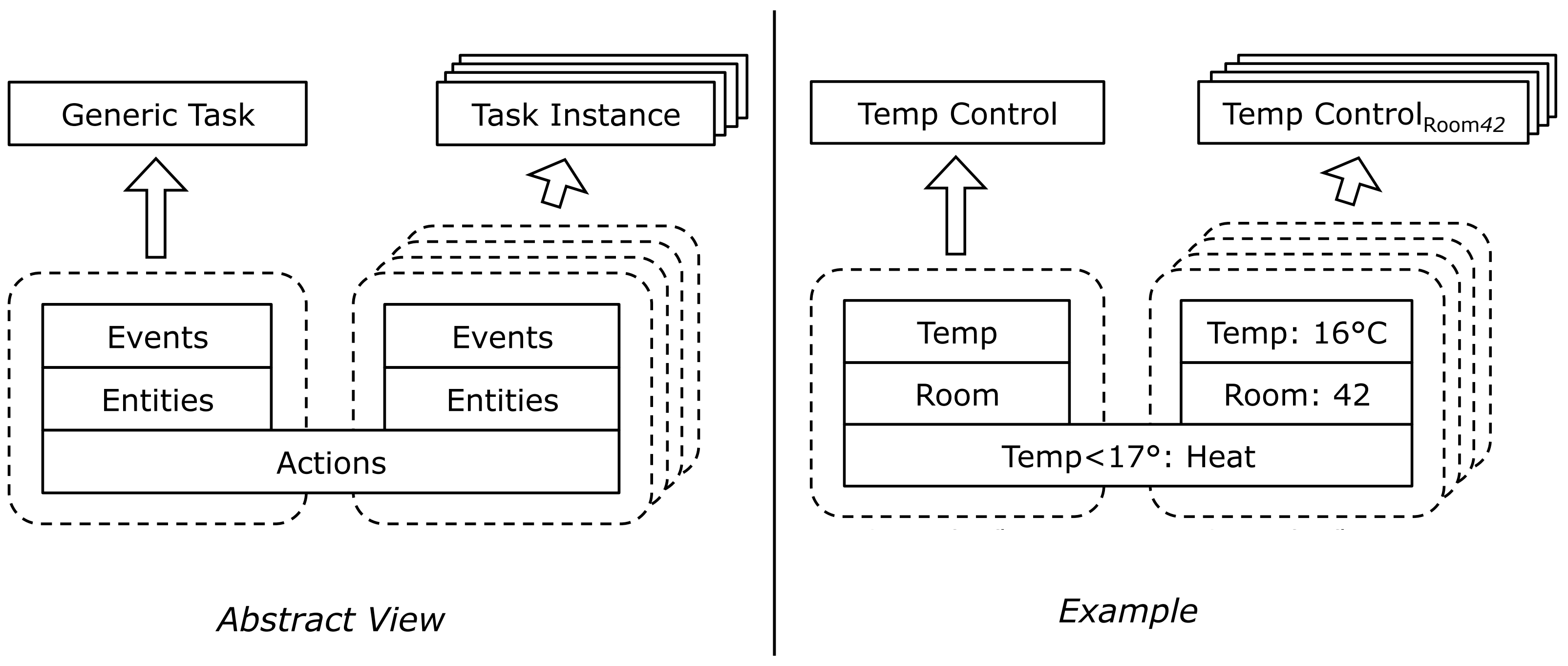
Reactive Applications

- Reactive applications integrate already available data (persistent data) with data just coming into existence (streams of events)
- Persistent data is accessed in a pull-based fashion:
  - Services query databases (request/reply)
  - Services are combined in service-oriented architectures (SOA)
- Streams of events are push-based:
  - Eventlets receive relevant events
  - Eventlets are combined in event-driven architectures (EDA)



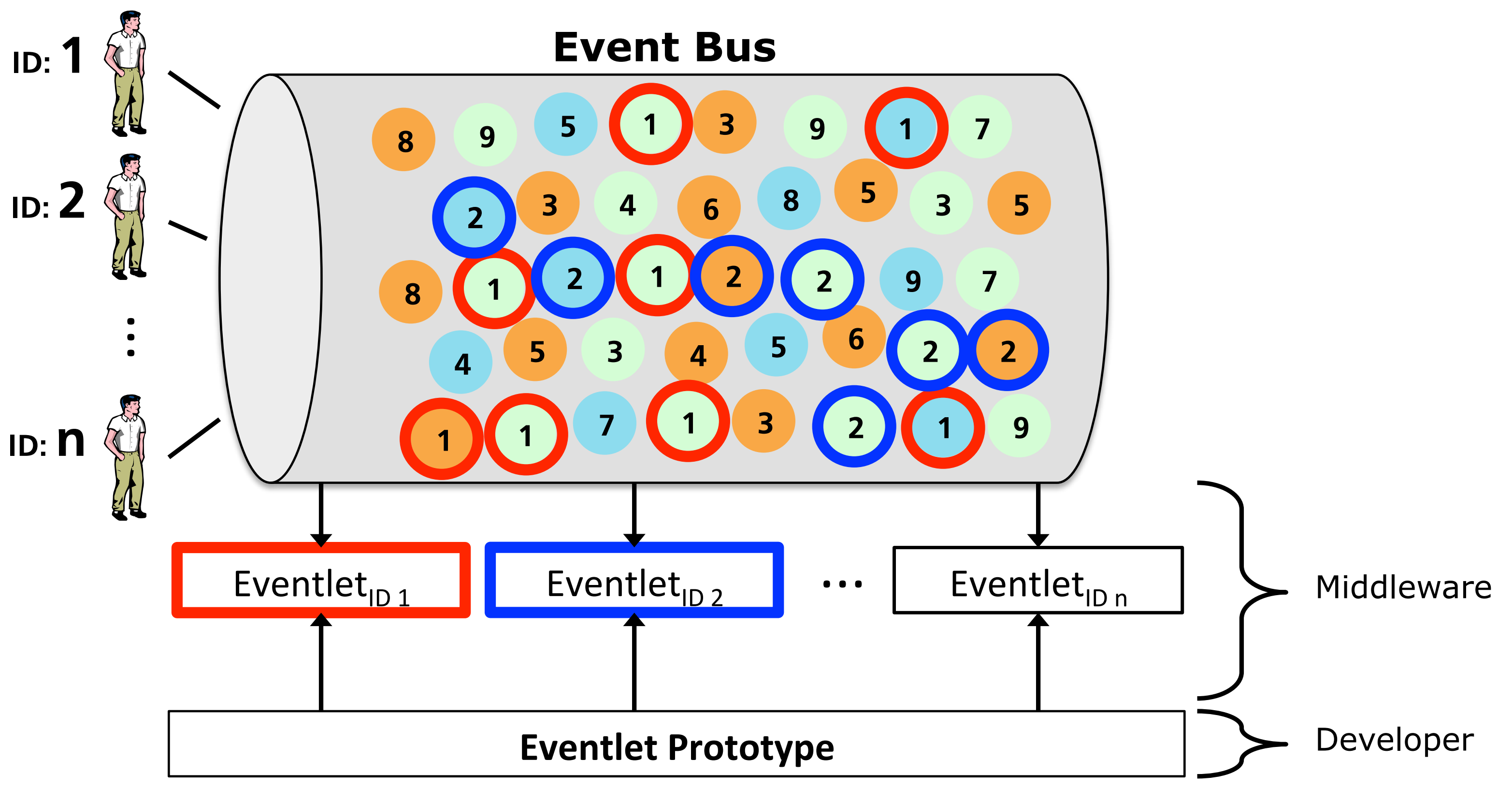
Generic Event-driven Tasks

- Eventlets are containers for generic event-driven tasks that encapsulate application logic invoked by events
- Tasks are generic in the sense that the same actions can apply to multiple entities and events, e.g., the application logic for a room temperature control can be reused for other rooms



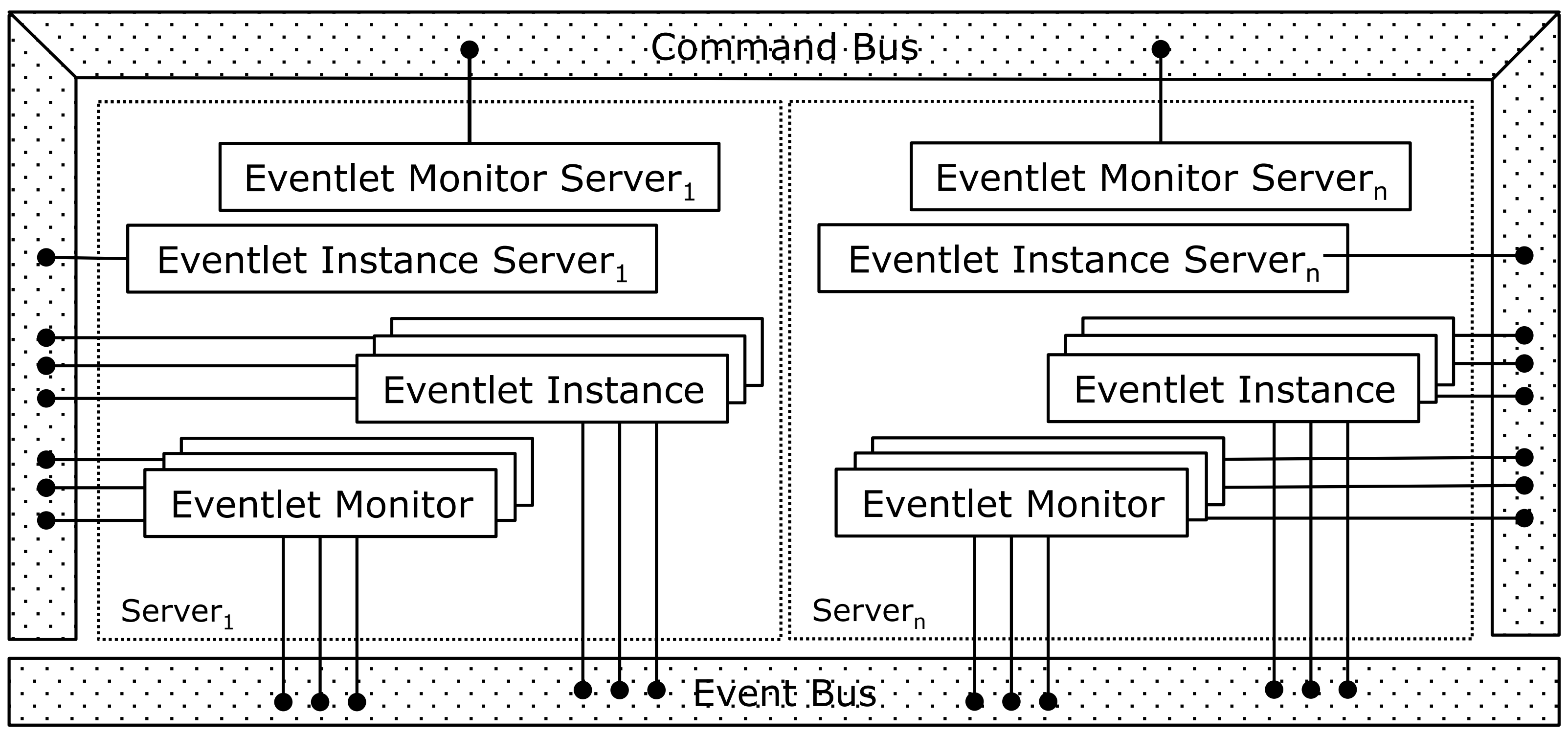
Eventlets

- Eventlets encapsulate application logic for groups of events that are associated with real world entities, e.g., persons
- Design paradigm
  - Write application logic for *"An Entity"* (Eventlet Prototype)
  - Eventlet middleware applies application logic for *"Each Entity"* when necessary (Eventlet Instance)



Eventlet Middleware

- Distributed framework for scalability
- Based upon Java Message Service (JMS)
- Registration of Eventlet prototype triggers creation of Eventlet monitor
- Eventlet monitors are responsible to create Eventlet instances
- Eventlet instances follow lifecycle: application logic upon instantiation, removal, expiration, and event arrival
- Asynchronous event handling for high performance


Command Bus
Eventlet Monitor Server$_1$
Eventlet Instance Server$_1$
Eventlet Instance
Eventlet Monitor
Server$_1$
Eventlet Monitor Server$_n$
Eventlet Instance Server$_n$
Eventlet Instance
Eventlet Monitor
Server$_n$
Event Bus

Contact: Stefan Appel, appel@dvs.tu-darmstadt.de | http://www.dvs.tu-darmstadt.de/