



Adventures in Real-Time Python NoSQL-style

Dr. Chris Zimmermann

Redis Labs

29. 7. 2021



- PhD in reflective operating system architectures
- First crush on Linux: kernel 0.95
- Tech support + more @ FraLUG
- Arch package maintainer
- Hobbies include:
 - SDLC
 - IT security and other forms of black art
 - Community liaison / solution architect @ redislabs



THE LINUX INLAWS
linux-inlaws.eu

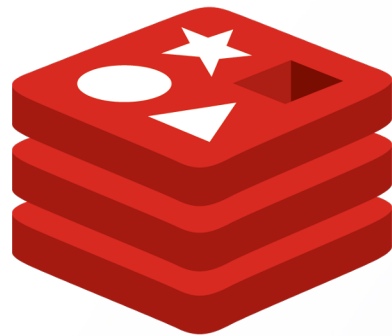
For Luca &
the CTO team @ Redis Labs

Agenda

- Redis Overview
- Multi-modal DB aspects
- RedisGears
- Demo
- Wrap-up / Q&A

Agenda

- *Redis Overview*
- Multi-modal DB aspects
- RedisGears
- Demo
- Wrap-up / Q&A



redis

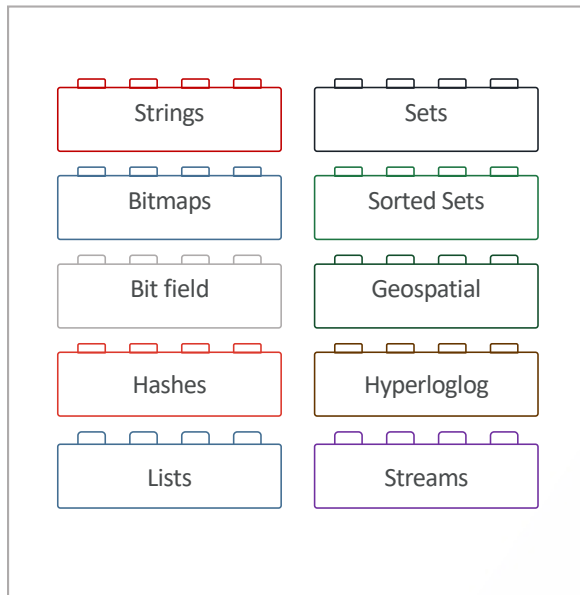
An **In-memory open source database**, supporting a variety of high performance operational, analytics or hybrid use cases

The Redis Community

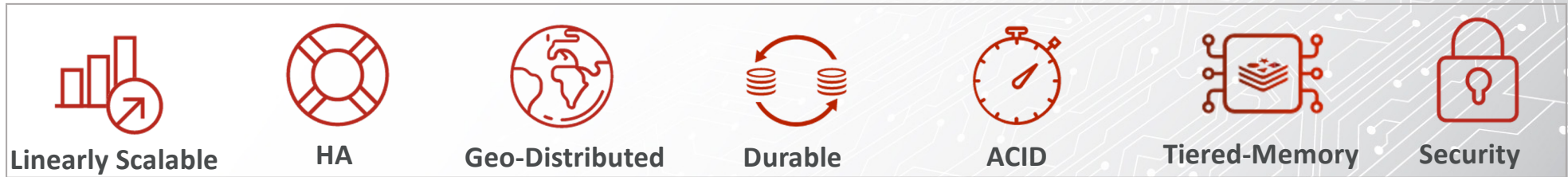
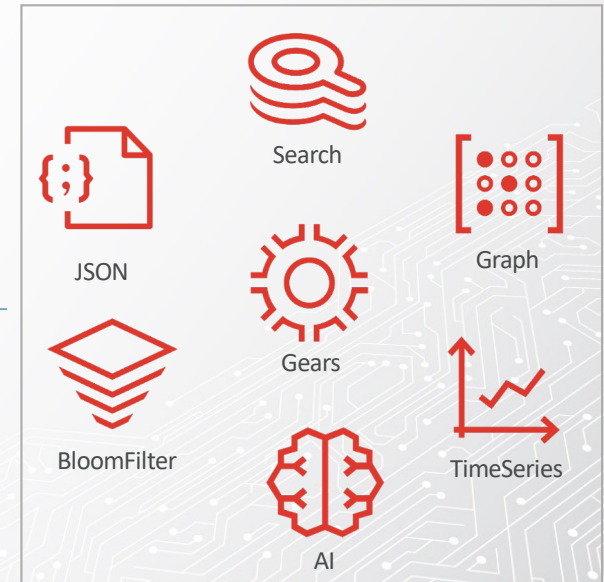


Redis and its ecosystem

Redis Datastructures



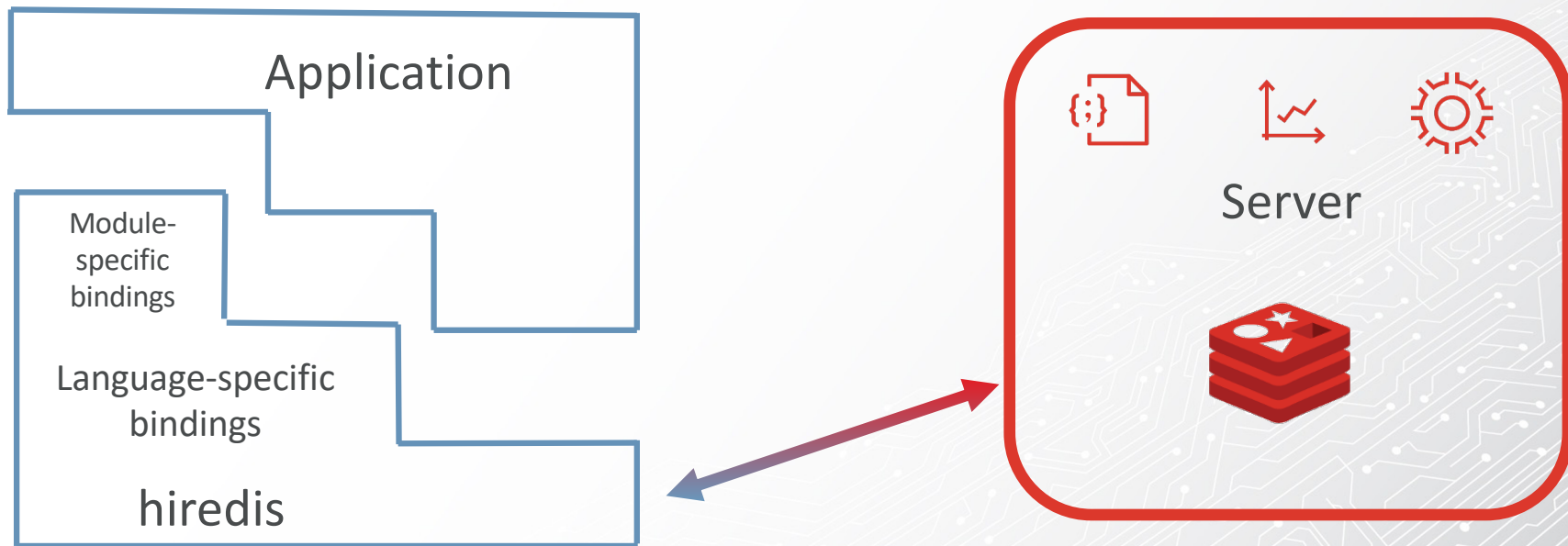
Redis Modules



Agenda

- Redis Overview
- *Multi-modal DB aspects*
- RedisGears
- Demo
- Wrap-up / Q&A

Modules – Principle Architecture



Module example: RedisTimeSeries



RedisTimeSeries

Main capabilities

- Downsampling/compaction
- Indexing, queries, aggregation
- Compression (double-delta encoding)
- Integration(Grafana, Prometheus)

Use cases

- Monitoring, filtering
- IoT

Module example: RedisAI



RedisAI

Main capabilities

- Combination of neural networks and redis for low-latency inference
- “Take the data to the model”
- New redis datatype: Tensor + Model
- Supported BPN backends:
 - TensorFlow, TensorFlow Lite
 - PyTorch
 - ONNX

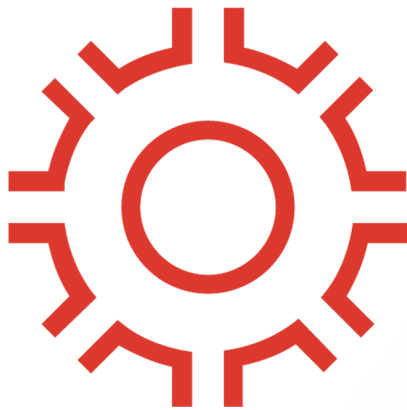
Use cases

- Real-time stack for Deep Learning & Machine Learning applications

Agenda

- Redis Overview
- Multi-modal DB aspects
- *RedisGears*
- Demo
- Wrap-up / Q&A

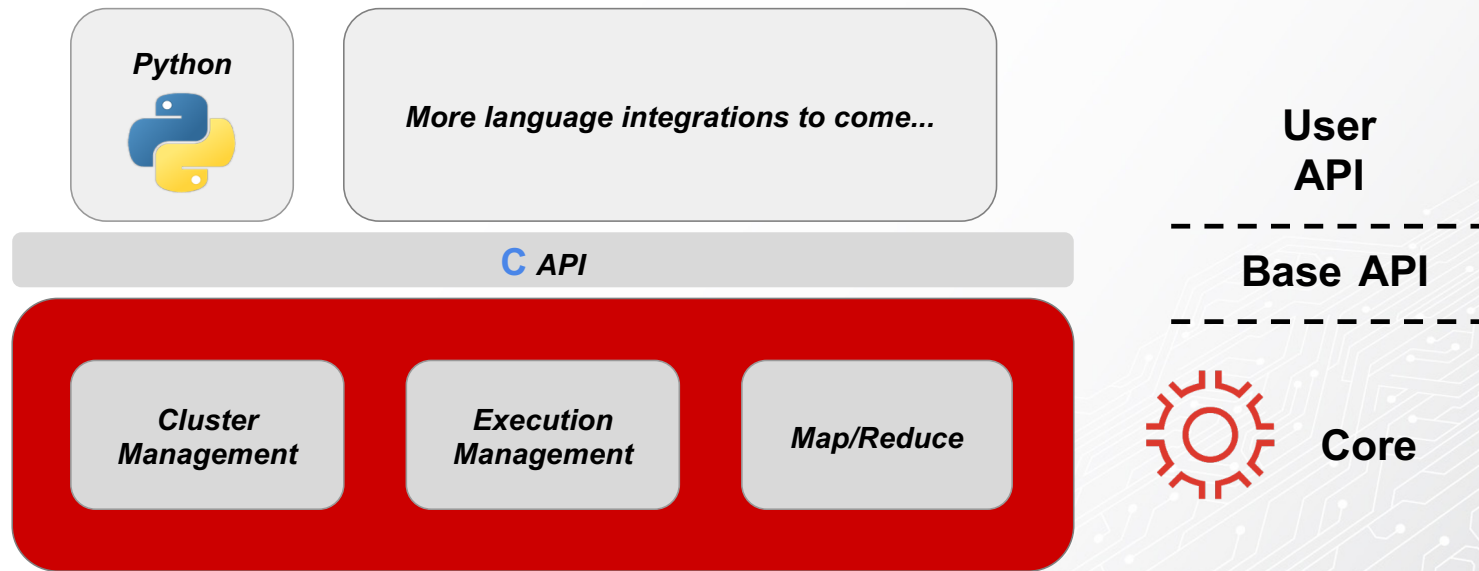
Redis Gears



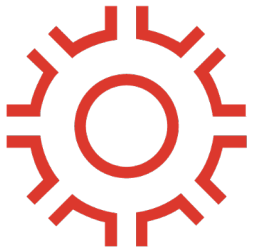
RedisGears:

- Serverless engine
- Operation: transactions, batches & events
- Dynamic framework for data flow implementation
- Abstraction layer for data distribution, clustering & deployment

Principal Architecture



Using RedisGears



```
gb = GearsBuilder()
gb.map(lambda x: x['value']) # turn records into sentences
gb.flatmap(lambda x: x.split(' ')) # split sentences to words
gb.countby() # count each word's occurrences
gb.run()
```

```
cat gear.py | redis-cli -x "RG.EXECUTE"
```

```
127.0.0.1:6379> set foo "test"
```

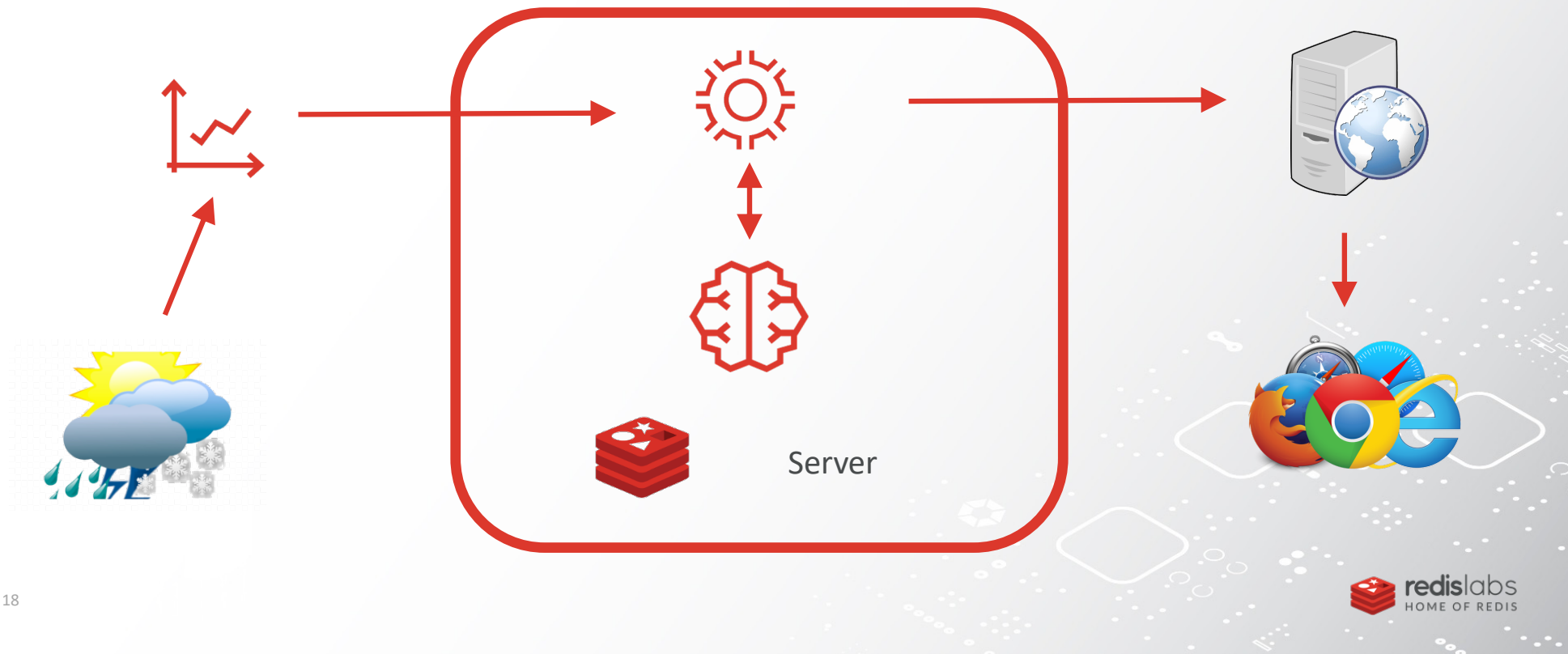
```
127.0.0.1:6379> set fool "this is a test"
```

```
1) 1) {"key": "test", "value": 2}
   2) {"key": "is", "value": 1}
   3) {"key": "a", "value": 1}
   4) {"key": "this", "value": 1}
```


Agenda

- Redis Overview
- Multi-modal DB aspects
- RedisGears
- *Demo*
- Wrap-up / Q&A

Demo: Time-Series Data Prediction



Further reading

- **redis:** `redis.io`
- **RedisGears:** `redisgears.io`
- **RedisAI:** `redisai.io`
- **TF timeseries forecasting:**
`https://www.tensorflow.org/tutorials/structured_data/time_series`
- **Redis Labs University:** `university.redislabs.com`

Q&A



Thank You!

Dr. Christoph Zimmermann
christoph@redislabs.com
+49 151 428-45087

