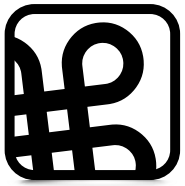


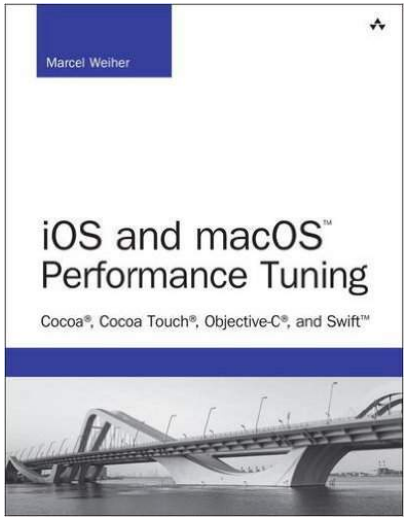
Macoun

Performance Architektur

Marcel Weiher

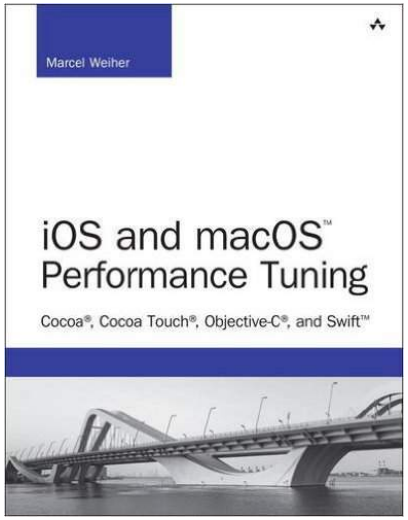
Performance Architektur

- Allgemeine Anmerkungen
- Call/Return
- Layering



Performance Architektur

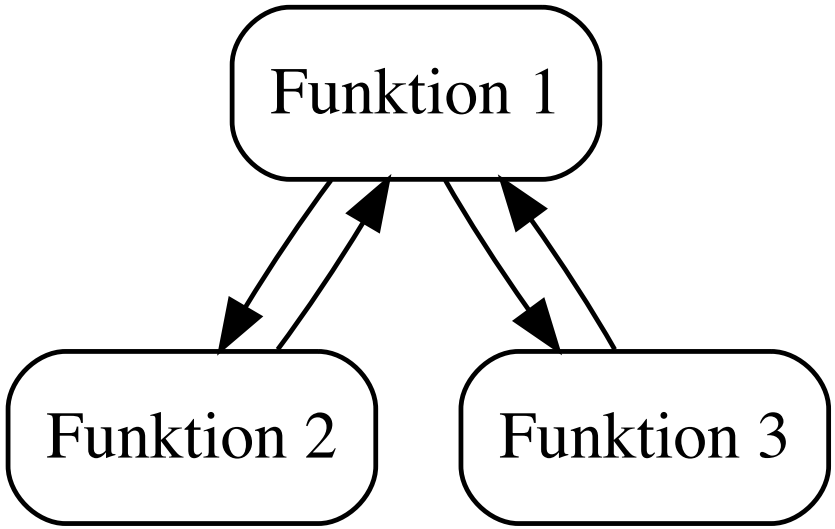
- **Allgemeine Anmerkungen**
- Call/Return
- Layering



Architektur?

- Grobgliederung eines Softwaresystems
- Komponenten + Konnektoren
- Architekturstil: Call/Return, Pipes+Filters, Event,…

Architekturstil: Call/Return



Architekturstil: Pipes+Filters

"We should forget about small efficiencies, say about 97% of the time,"

"Premature optimization is the root of all evil."

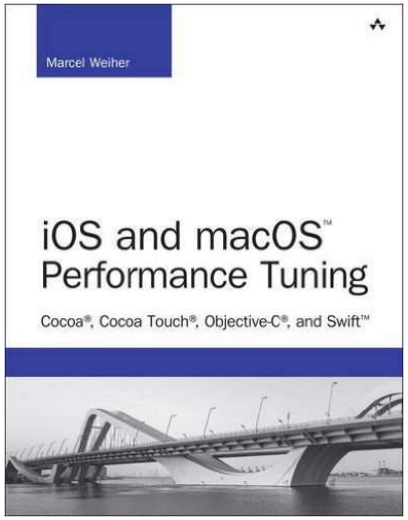
"Yet we should not pass up our opportunities in that critical 3%."

Performance Architektur

- Performance “im Großen”
- Optimierbarkeit
- Apple: Tiger / Leopard

Performance Architektur

- Allgemeine Anmerkungen
- **Call/Return**
- Layering



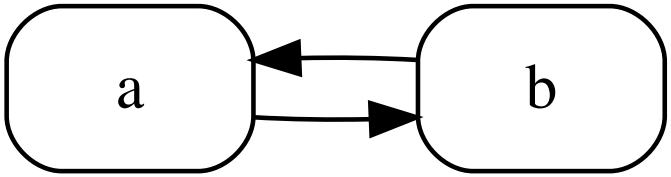
Call/Return Architekturstil

- Prozedural, OO, FP, überall
- Nicht unproblematisch…

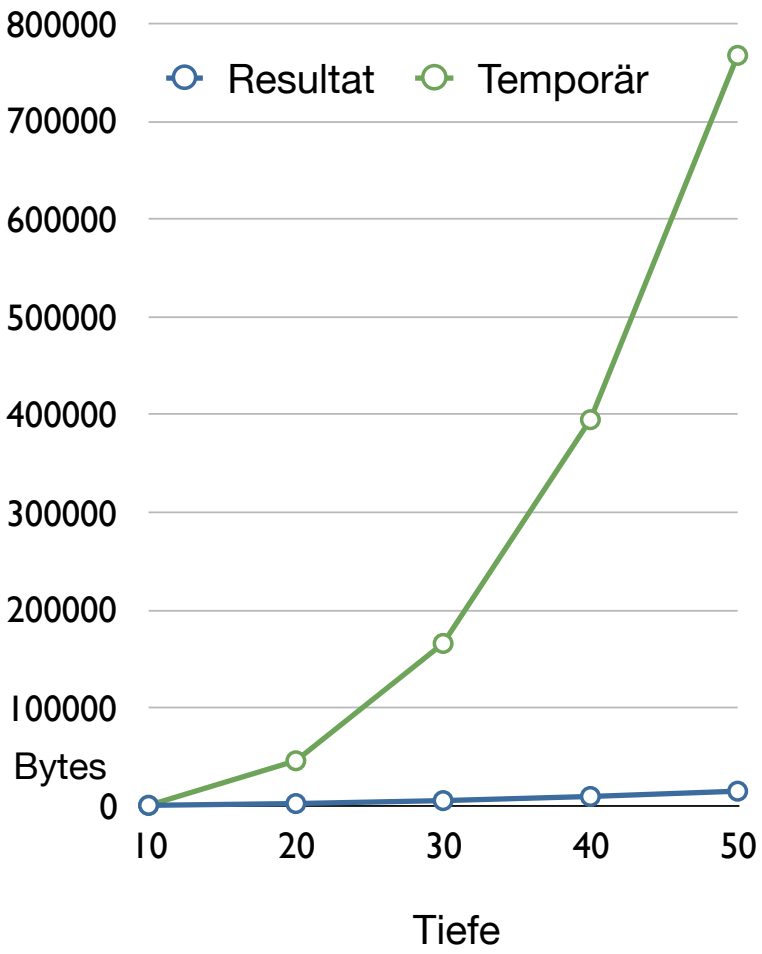
+ `description`

    Returns a string that represents the contents of the receiving class.

Demo 1

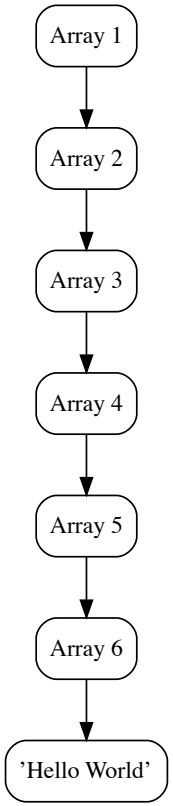

a
b

Demo 2


800000
700000
600000
500000
400000
300000
200000
100000
Bytes
0
Resultat
Temporär
10
20
30
40
50
Tiefe

Call/Return Architekturstil

- Speicherbedarf O(Breite * Tiefe) → $O(n^2)$
- Zeitaufwand ~ Speicherbedarf
- 2000 Elemente: 3.6 Gigabyte, 2 Sekunden



((((((“Hello World”))))))

(((((“Hello World”)))))

((((“Hello World”))))

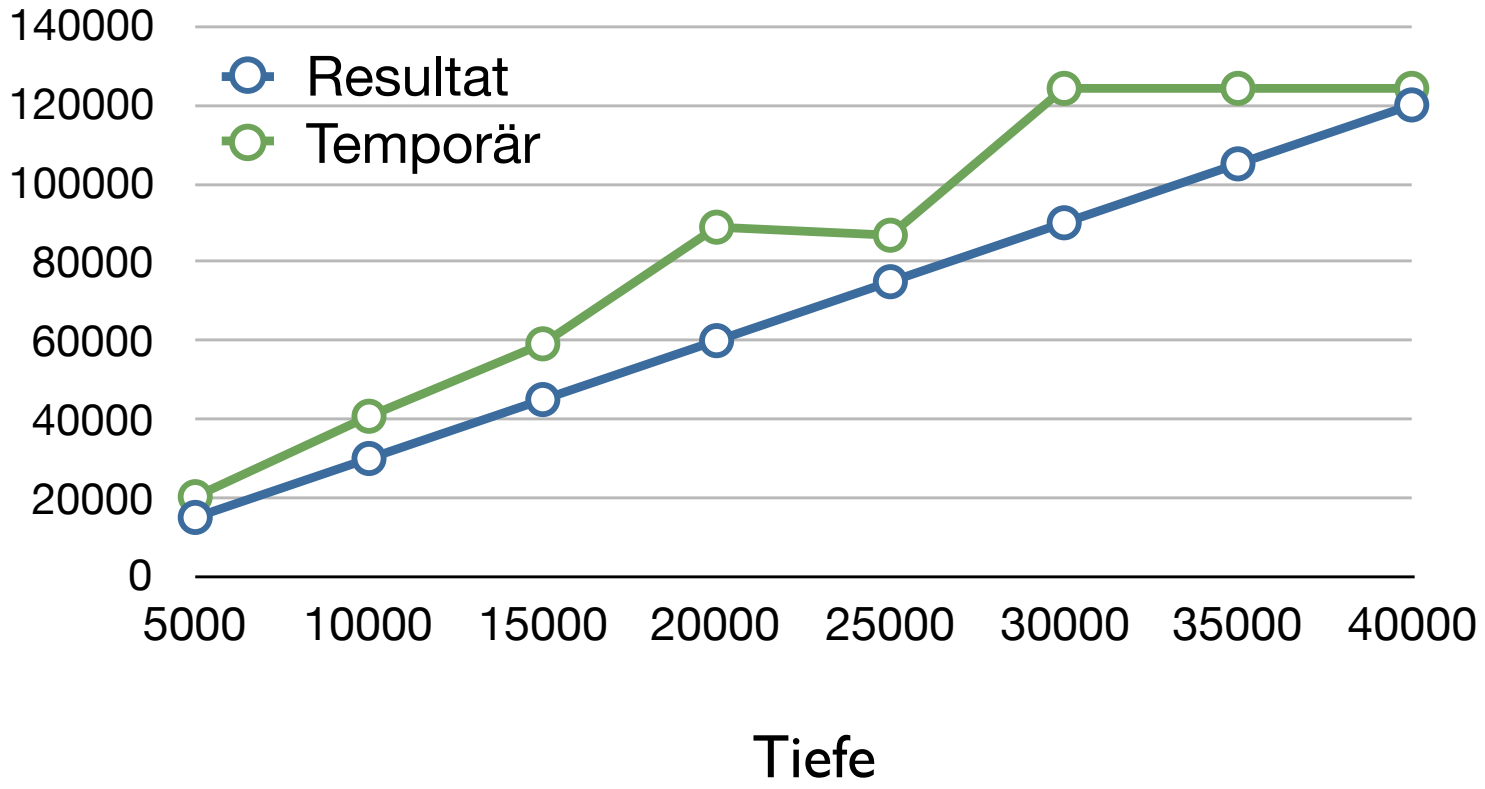
(((“Hello World”)))

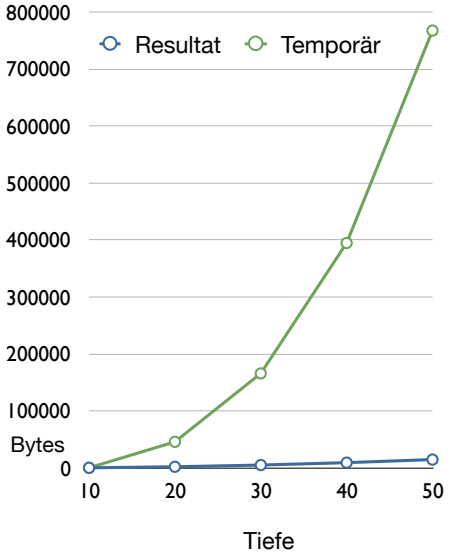
((“Hello World”))

(“Hello World”)

“Hello World”

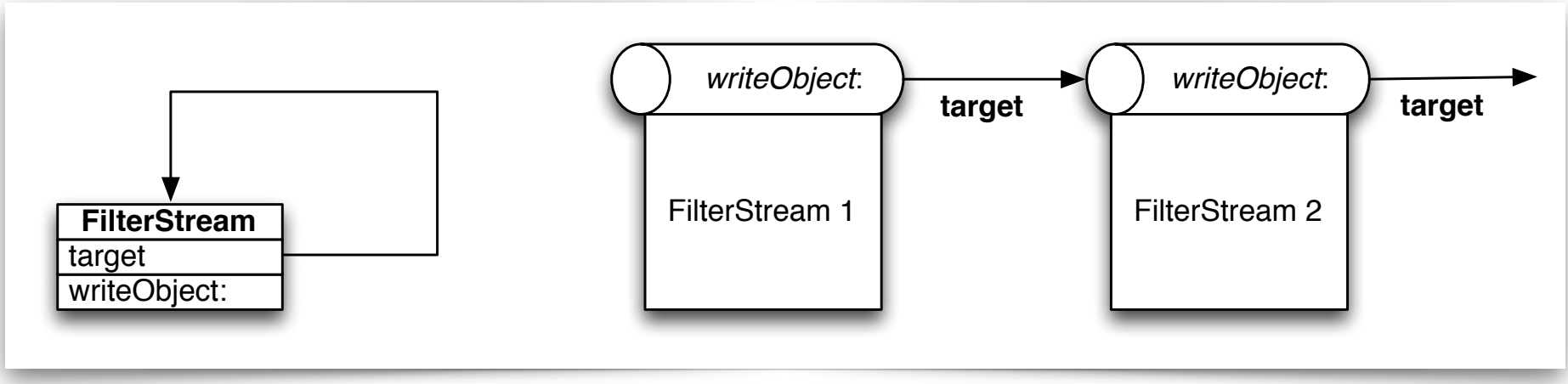
Demo 3

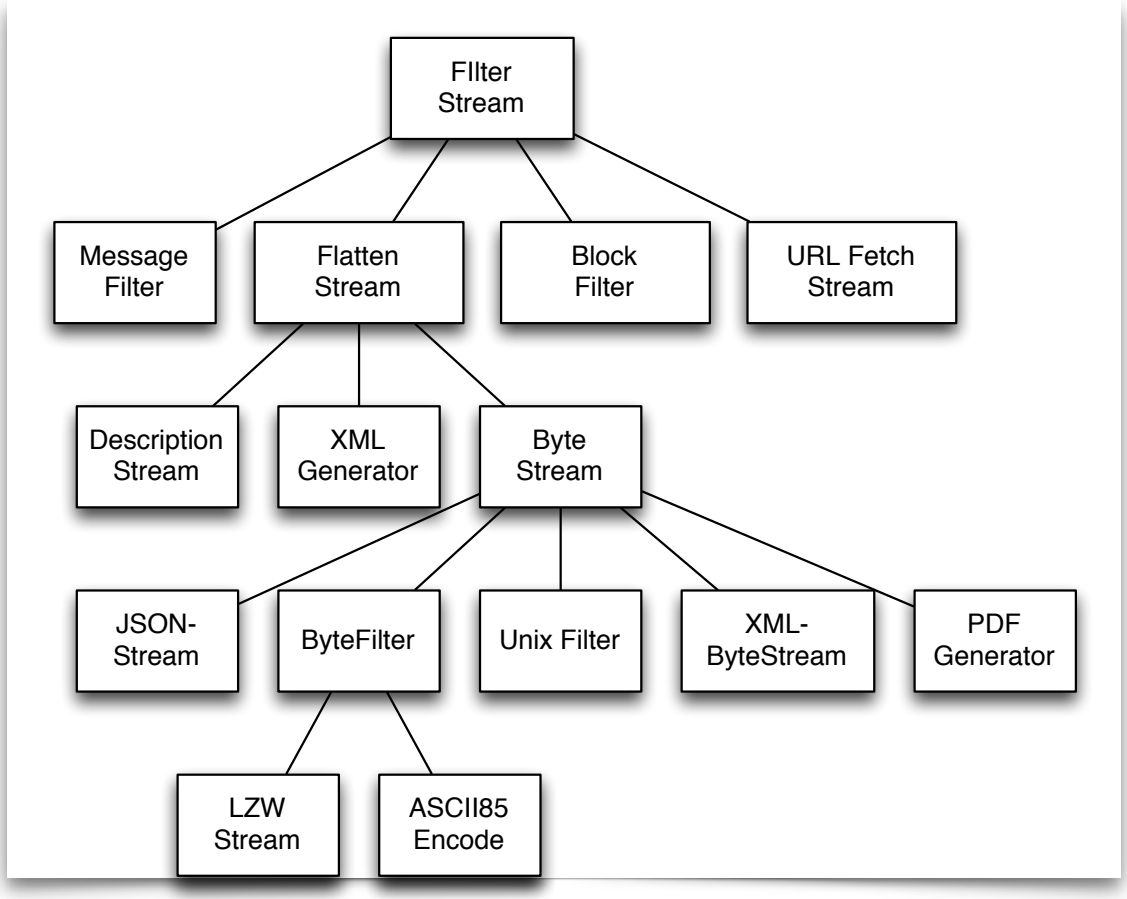

140000
120000
100000
80000
60000
40000
20000
0
Resultat
Temporär
5000
10000
15000
20000
25000
30000
35000
40000
Tiefe


800000
700000
600000
500000
400000
300000
200000
100000
Bytes
0
Resultat
Temporär
10
20
30
40
50
Tiefe

Pipe+Filter Architekturstil

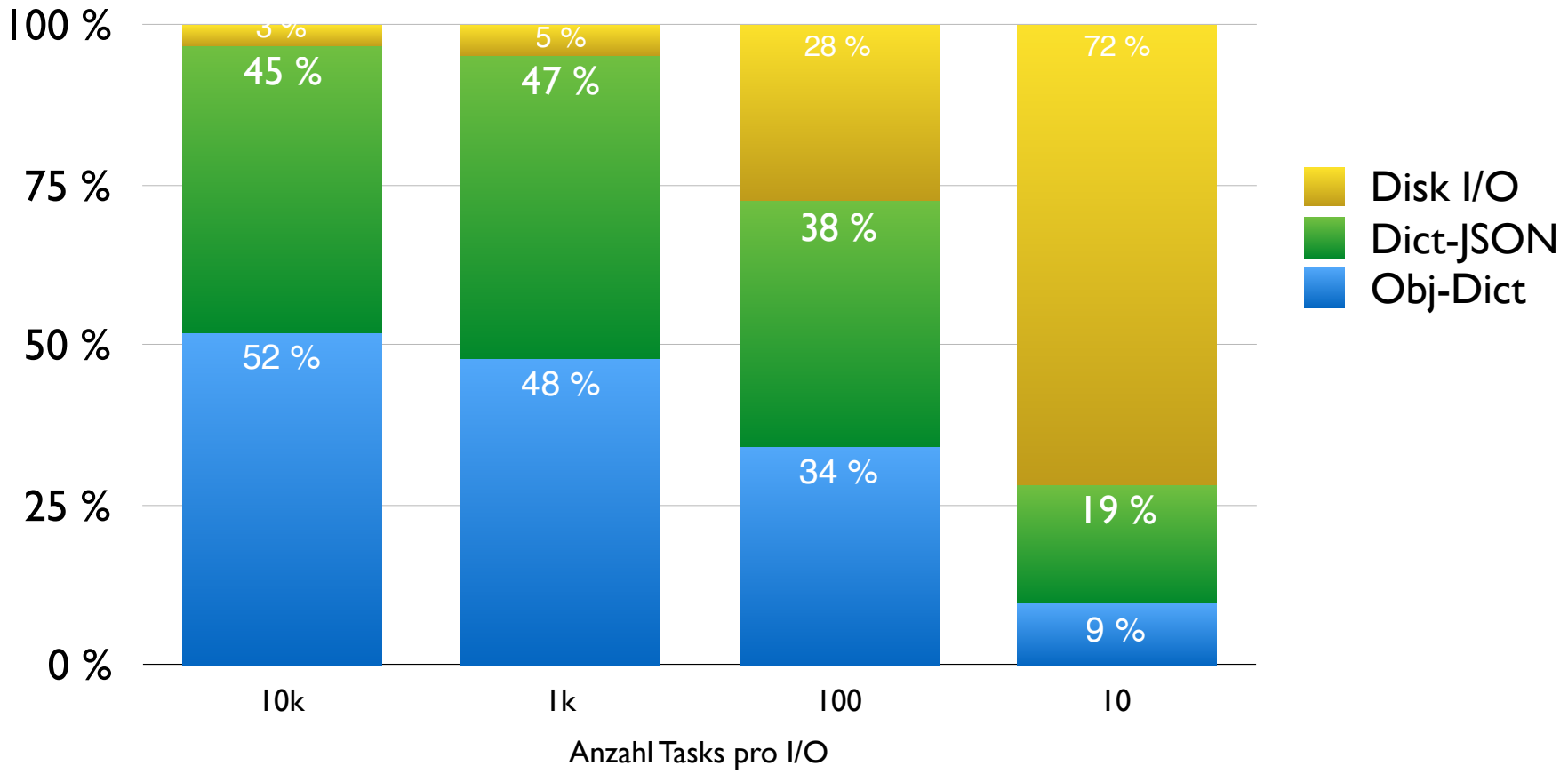
- Inkrementell / linear / komponierbar
- 80000 Elemente: 290 Kilobyte, 70 ms (~ 5.8 TB, 1h)

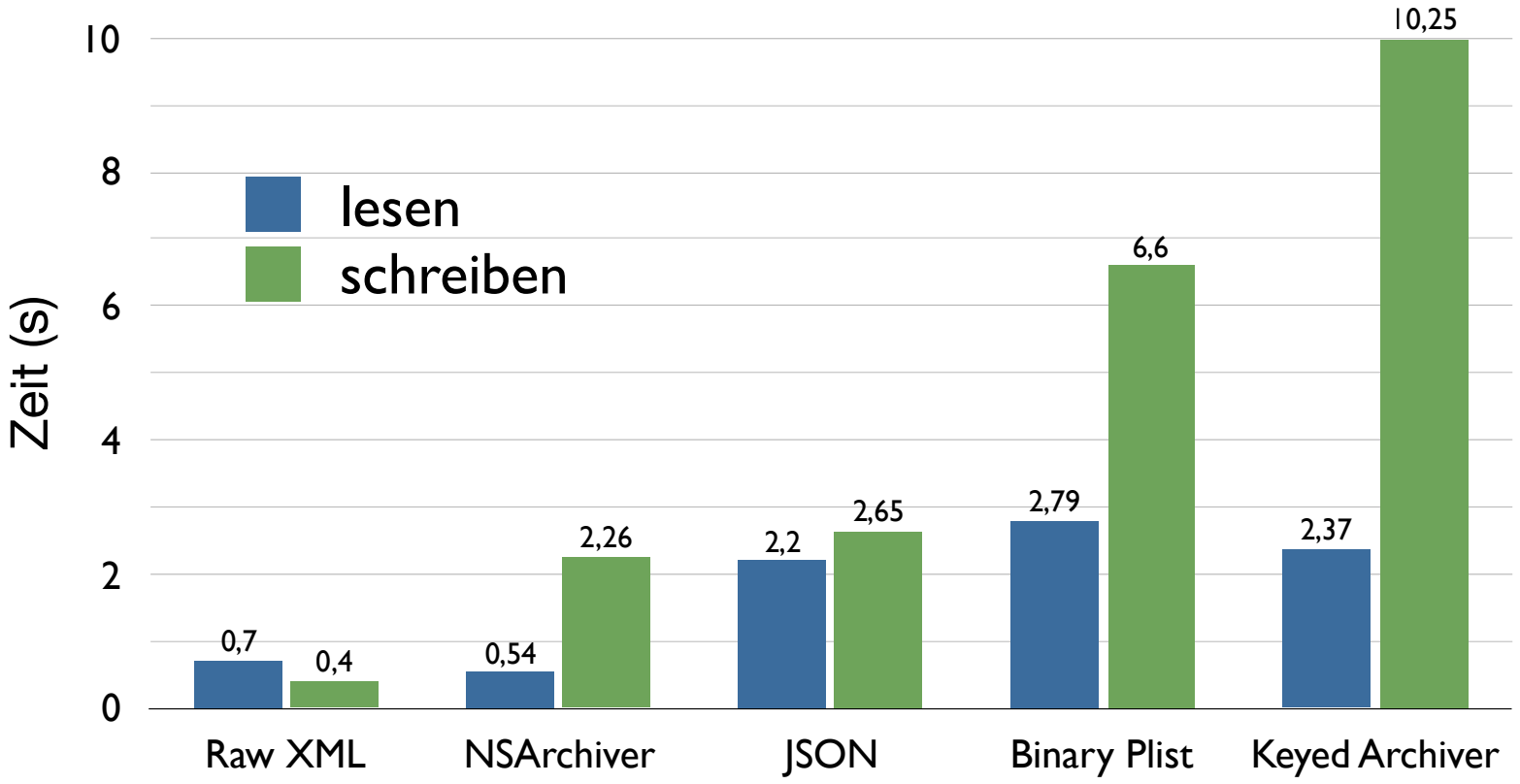

FilterStream
target
writeObject:
writeObject:
target
FilterStream 1
writeObject:
target
FilterStream 2

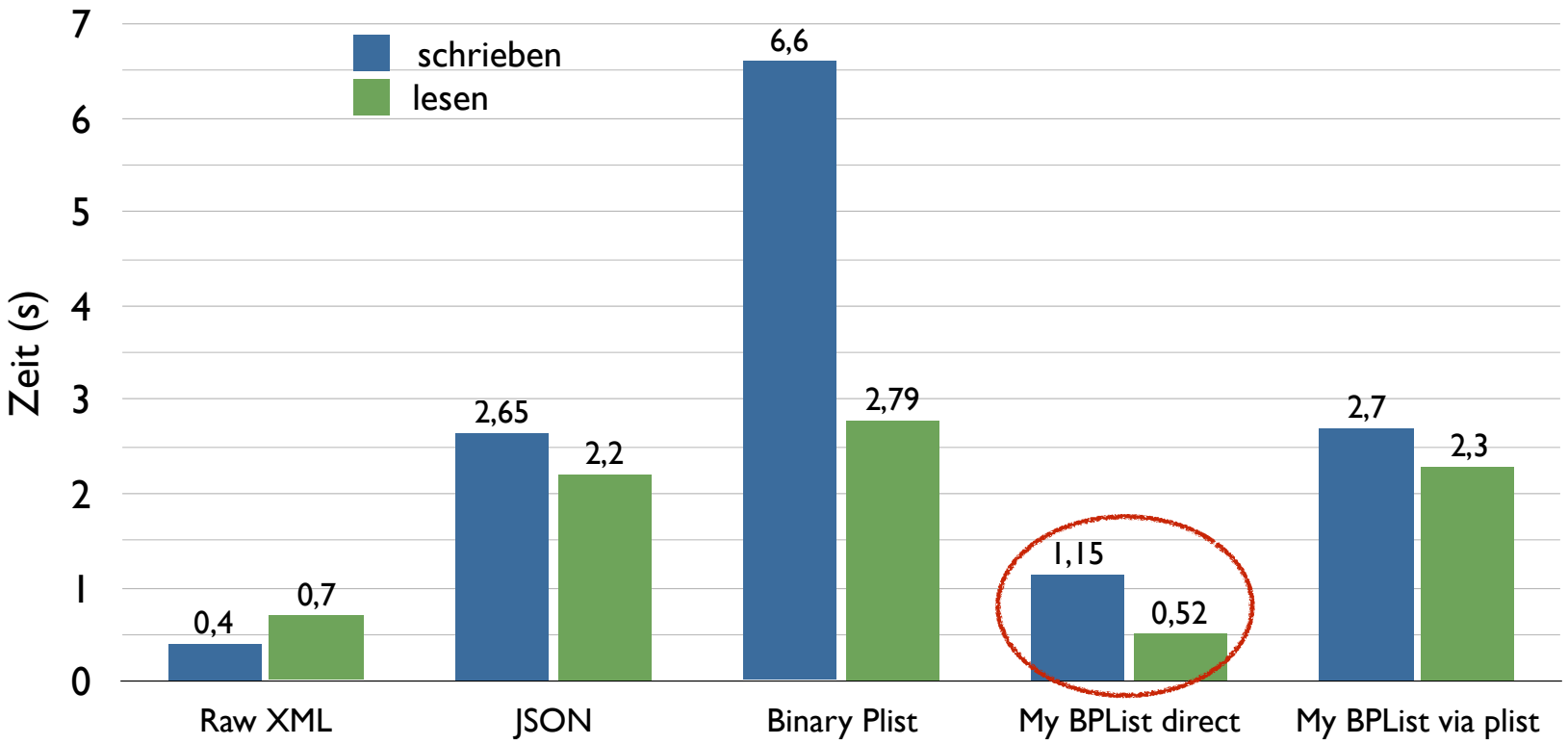

FIlter Stream
Message Filter
Flatten Stream
Block Filter
URL Fetch Stream
Description Stream
XML Generator
Byte Stream
JSON-Stream
ByteFilter
Unix Filter
XML-ByteStream
PDF Generator
LZW Stream
ASCII85 Encode

Na und?

- NS*Serialization APIs haben das gleiche API Problem
- Generische Zwischenrepresentation (“P-List”)

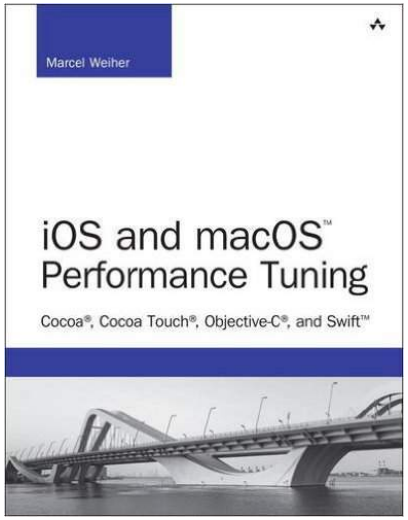

100 %
75 %
50 %
25 %
0 %
3 %
45 %
52 %
5 %
47 %
48 %
28 %
38 %
34 %
72 %
19 %
9 %
10k
1k
100
10
Anzahl Tasks pro I/O
Disk I/O
Dict-JSON
Obj-Dict


Zeit (s)
10
8
6
4
2
0
lesen
schreiben
0,7
0,4
0,54
2,26
2,2
2,65
2,79
6,6
2,37
10,25
Raw XML
NSArchiver
JSON
Binary Plist
Keyed Archiver

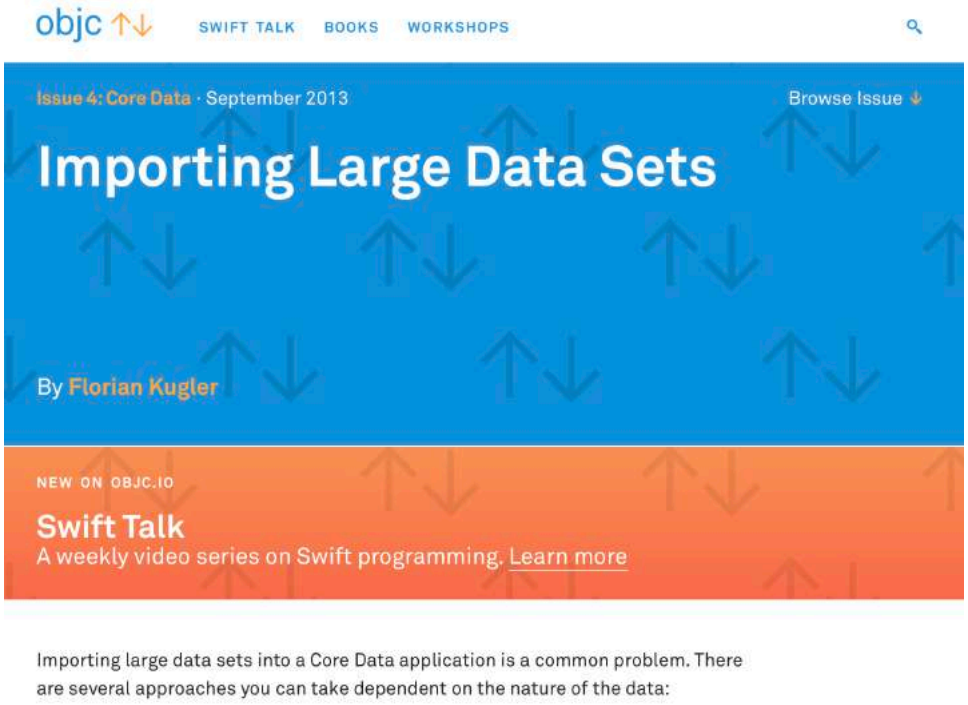

Zeit (s)
7
6
5
4
3
2
1
0
schrieben
lesen
0,4
0,7
2,65
2,2
6,6
2,79
1,15
0,52
2,7
2,3
Raw XML
JSON
Binary Plist
My BPList direct
My BPList via plist

Performance Architektur

- Allgemeine Anmerkungen
- Call/Return
- **Layering**

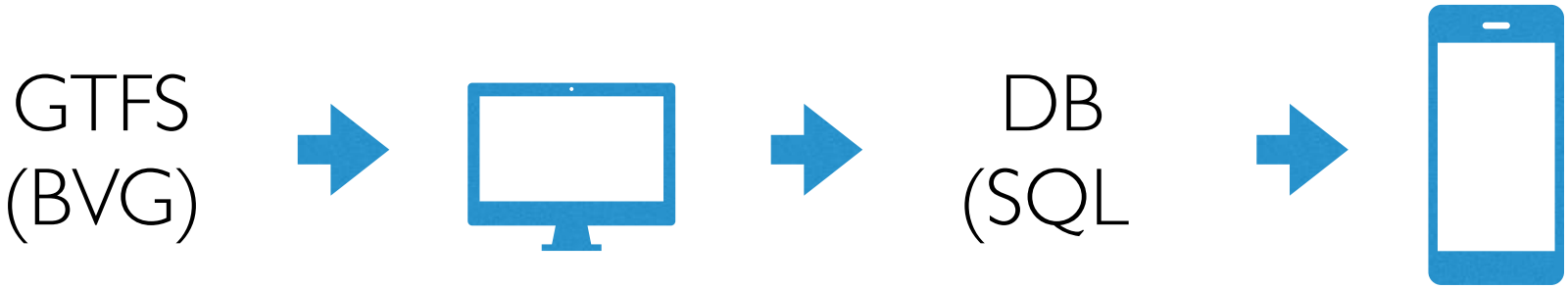


Fahrplandaten

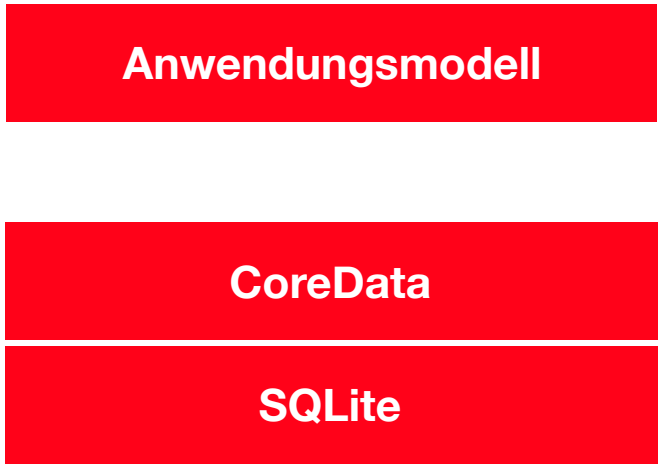


Used with Permission

Fahrplandaten



Fahrplandaten: Schichten



Ergebnis mit Schichten

- ca. 120MB CSV → SQLLite DB
- Import: 22 Minuten
- Abfrage auf Gerät: 500 ms

Fahrplandaten: Ports/Adapters

- Von drinne noh drusse
- Fokus ist das Modell
- Aussenwelt nur über Ports

Fahrplan: GTFS Daten (CSV)

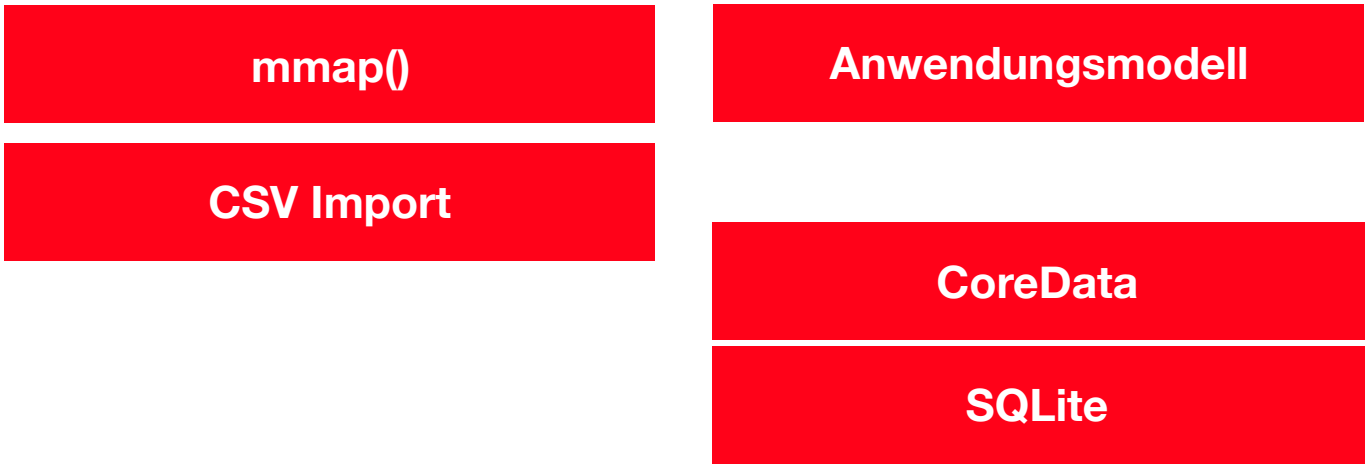
```
stop_id,stop_code,stop_name,stop_desc,stop_lat,stop_lon,...
9003101,,U Hansaplatz (Berlin),,52.5181110,13.3421650,...
9003102,,S Bellevue (Berlin),,52.5197640,13.3469160,...
9003103,,S Tiergarten (Berlin),,52.5144000,13.3364690,...
9003104,,U Turmstr. (Berlin),,52.5258370,13.3424010,...
...
trip_id,arrival_time,departure_time,stop_id,stop_sequence,stop_headsign,...
1,18:16:00,18:16:00,9096310,...
1,18:18:00,18:18:00,9096364,...
1,18:19:00,18:19:00,9096311,...
1,18:21:00,18:21:00,9096365,...
```

Demo 4

Fahrplandaten: Ports/Adapters

- ca. 120MB CSV → 12 MB Binärdaten
- Import: 1 Sekunde (1000x)
- Abfrage auf Gerät: 58 µs (8000x)

Fahrplandaten: Ports/Adapters

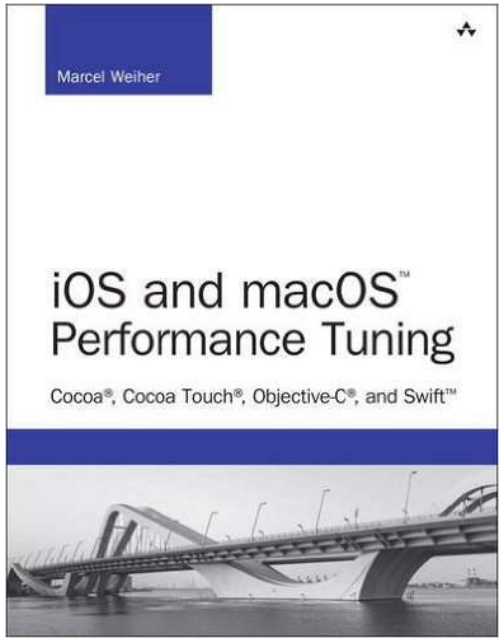


Performance Architektur

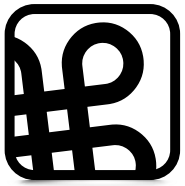
Marcel Weiher

Fragen?

- Performance “im Großen”
- Optimierbarkeit
- Call/Return → Pipe+Filter
- Schichten → Ports+Adaptors
- software architecture patterns shaw



Vielen Dank



Macoun