



# Profiling Flash

flexughh, 16.11.2010

# Agenda

- ① Profiling in Flash
- ② FDT 4 Profiler
- ③ Flash Player Internals
- ④ Beispiele
- ⑤ Tipps & FAQ

# Was ist Profiling?

dynamisches Analyseverfahren

Memory Profiling (Analyse des Speicherbedarfs)

Performance Profiling (Dauer und Häufigkeit eines Methodenaufrufs)

# Wie funktioniert Profiling?

Instrumentierung (Veränderung) der Software

Quellcode-Manipulation oder Bytecode-Injizierung

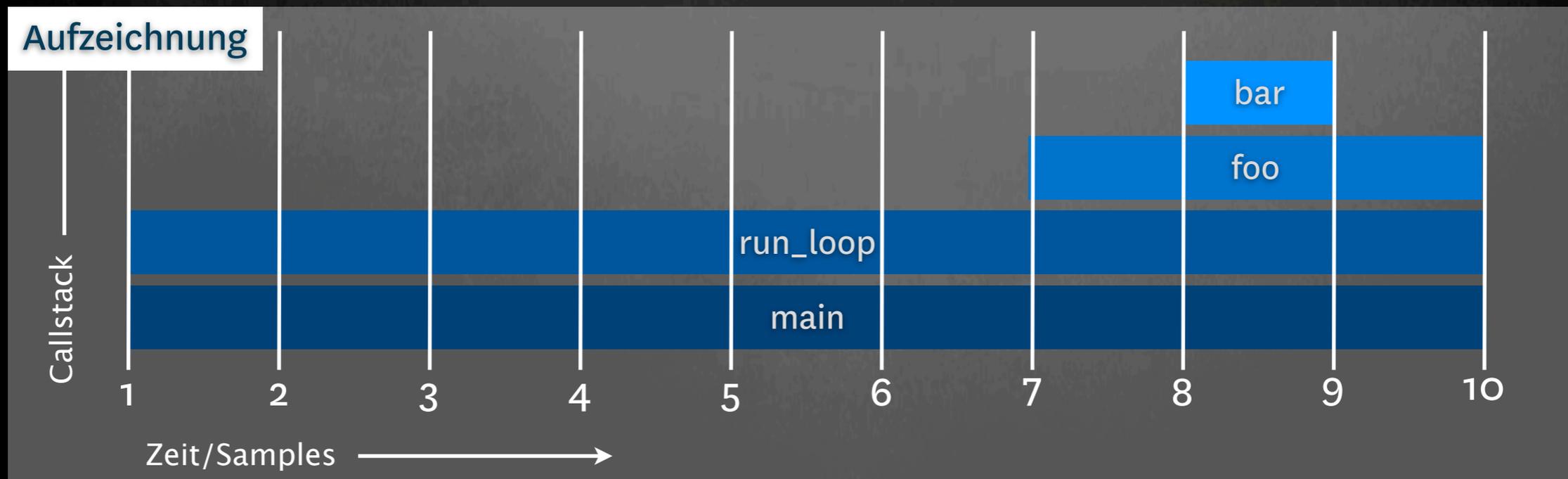
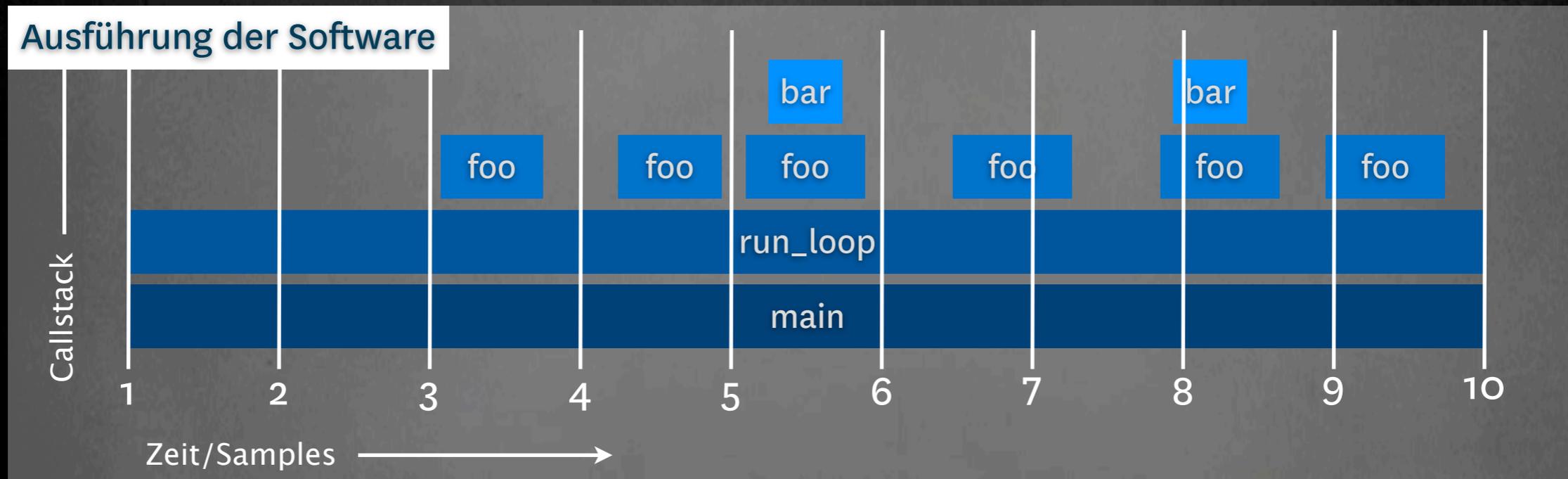
vollständige Daten zur Ausführung der Software

Sampling

statistisches Verfahren

Speichernutzung wird in festgelegtem Intervall aufgezeichnet

# Sampling



# Sampling in Flash

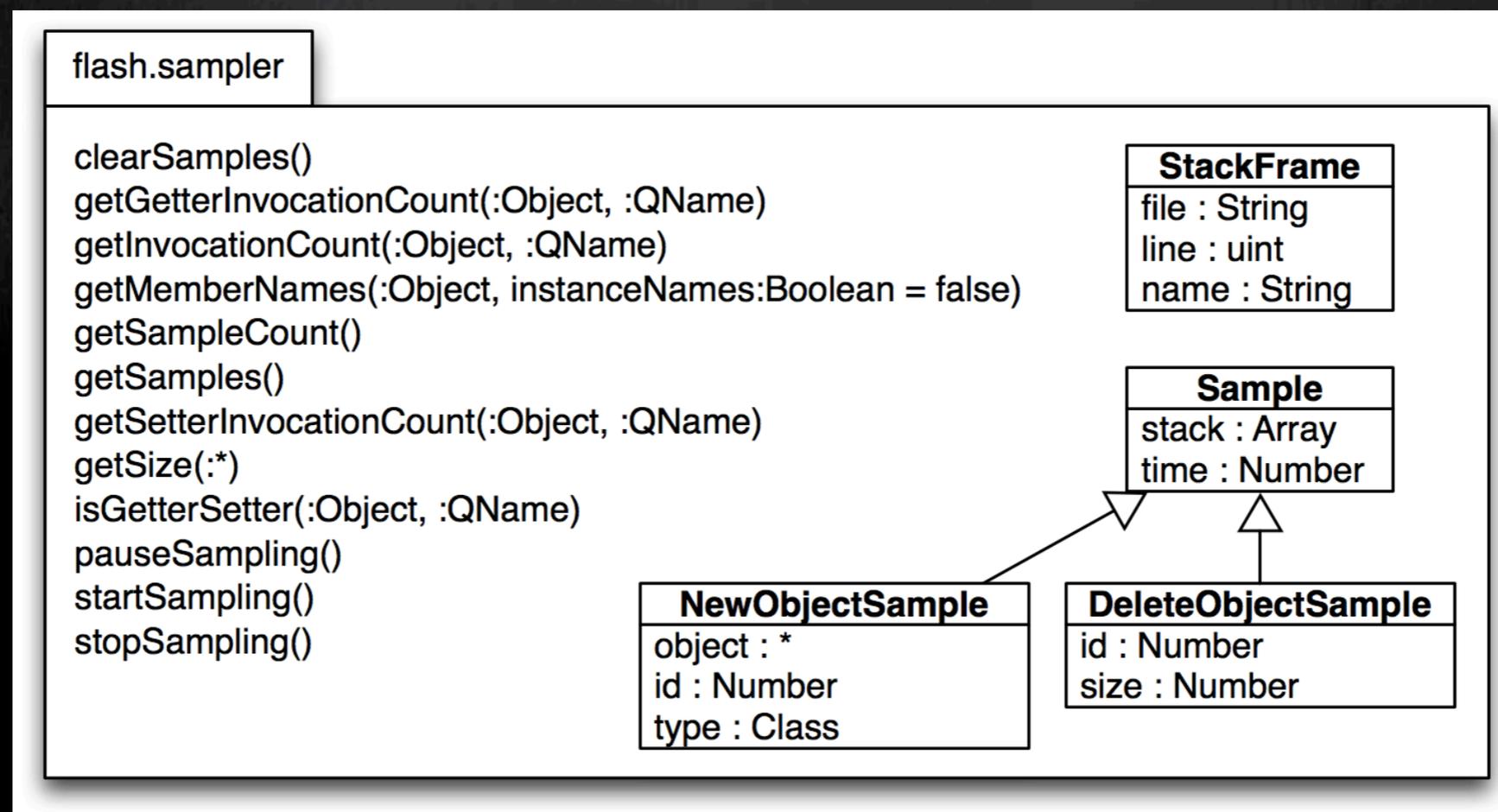
einzigste Möglichkeit in Flash (Intervall 1ms)

Aufzeichnung und Analyse erzeugt viel Mehraufwand

Ergebnisse unterscheiden sich (Ausführung, Plattform, Flash Player, Player Version)

nur Informationen über erzeugte Instanzen

High-Level API (flash.sampler Package)



# Warum wird oft auf Profiling verzichtet?

- ~~1. Performance-Probleme treten selten auf~~
- ~~2. Hardware ist schnell und günstig~~
- ~~3. Es ist zu teuer, reaktionsschnelle Software zu entwickeln~~
- ~~4. Wenn nötig, kann man die Software später „tunen“~~
- ~~5. Effizienz erfordert „tricky Code“~~

**Performance Analyse gehört zur täglichen Arbeit!**

# DEMO

# Flash Player Memory Management

GC wird nur bei Bedarf getriggert

Zwei phasiger Algorithmus

Mark

Ermittlung des Reference Count für jedes Objekt (nicht exakt)

ausgehend von allen Root-Elementen (Stage, ApplicationDomain, statische Definitionen, lokale Variablen)

wird inkrementell ausgeführt

Sweep

Speicherplatz nicht referenziertes Objekte wird freigegeben

Freier Speicher wird nicht unbedingt an das System zurückgegeben

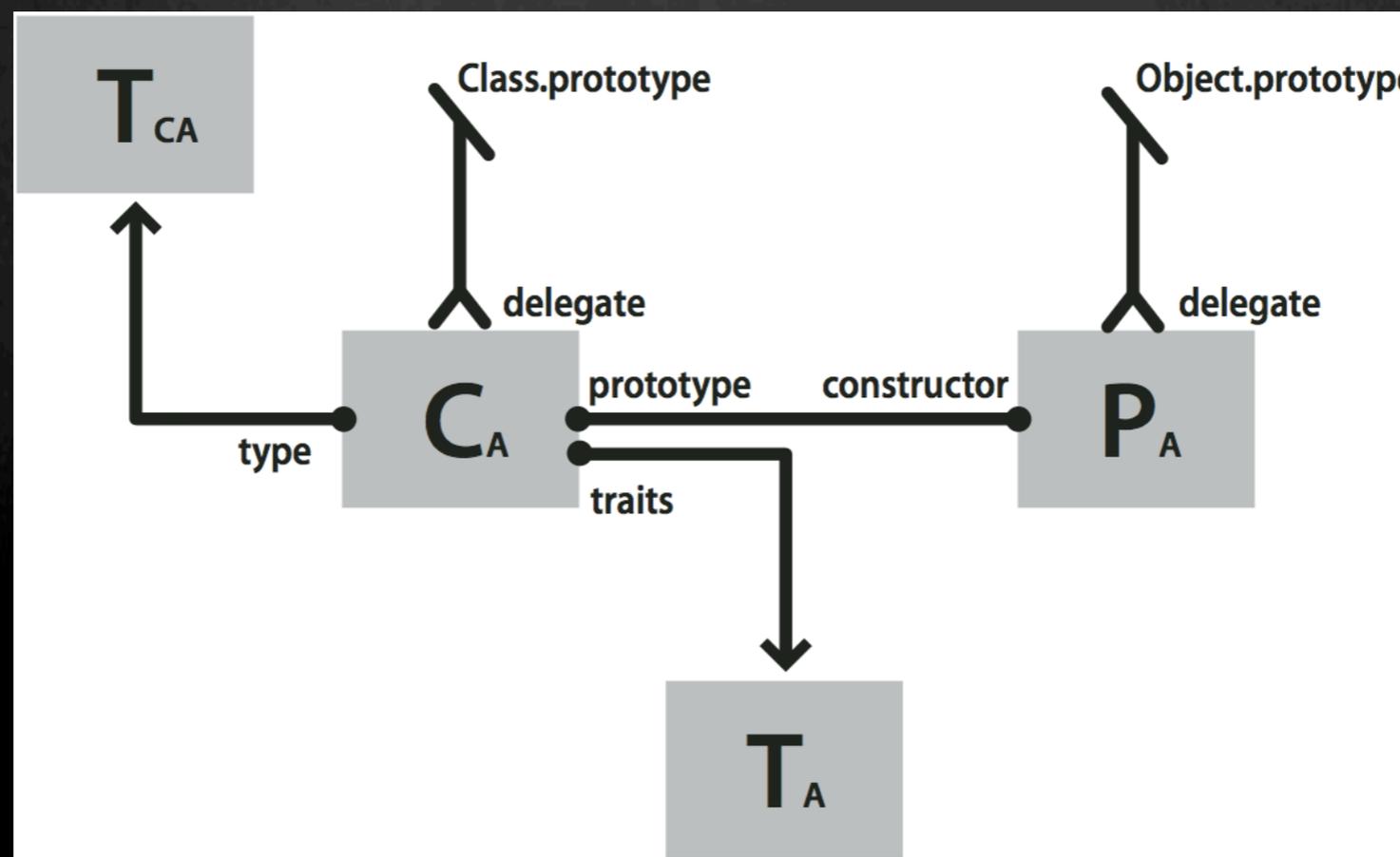
Übergangslagerung dereferenzierter Objekte und spätere Speicherfreigabe (reap)

# Objekt-Typisierung

Klassen in AS3 besitzen traits (Container für Objektbeschreibungen)

Eigenschaften typisierte Objekte können schneller aufgerufen werden

Traits ermöglichen optimierten Maschinencode (JIT)



DEMO

# Tipps

stetiger Einsatz während der Entwicklung (Architektur-Performance + Lerneffekt)

Memory + Performance Profiling beeinflussen sich gegenseitig

Kritische Bereiche isoliert testen

Berechnungen zusammenfassen

Bedingungen nach Erfüllungshäufigkeit sortieren

Bedingung für Schleifen nach Möglichkeit „auspacken“

# FAQ

POWERFLASHER GMBH  
BELVEDERALLEE 5  
52070 AACHEN

[www.powerflasher.de](http://www.powerflasher.de)

---

Meinhard Gredig  
Software Developer

E [mgd@powerflasher.de](mailto:mgd@powerflasher.de)

T 0241 91 88 0 - 237