Raum-Zeit-Constraints als Programmierabstraktion für Anwendungen in mobilen verteilten Systemen

Martin Däumler Dirk Müller Matthias Werner

Lehrstuhl Betriebssysteme
Fakultät für Informatik
Technische Universität Chemnitz

19. November 2009

Inhaltsverzeichnis

Motivation

Generische Unterstützung für Anwendungen in mobilen verteilten Systemen durch:

1. Abstraktion der Verteiltheit eines verteilten System (*System-Sicht*)
2. Explizite Behandlung von räumlichen, zeitlichen und bewegungsspezifischen Sachverhalten innerhalb einer Anwendung (*reale Raum-Zeit*)

- Arbeitstitel: **FlockOS**
  (**F**ederation of **l**inked **o**bjects with **c**ommon tas**k**s **O**perating **S**ystem)

Anwendung: Kartographie mittels kooperierender Satelliten

- Knoten-Sicht: Explizite Programmierung der Migration der Anwendung sowie der Kommunikation der Daten zwischen den Satelliten
- System-Sicht: Programmierung einer Gesamtanwendung, die die Beobachtung eines räumlichen Gebiets beschreibt

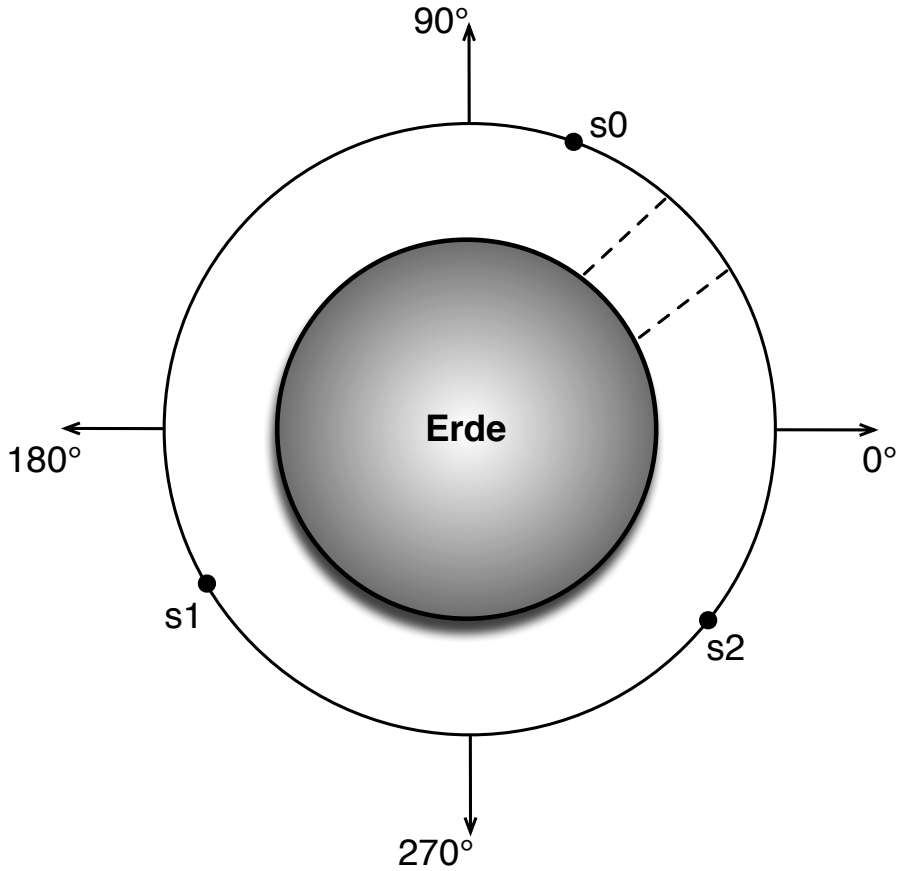


Abbildung: Anwendung Kartographie mittels Satellitensystem

Anwendung: Roboterfußball

- Knoten-Sicht: Alternative Realisierungen eines Spielzuges müssen explizit programmiert werden um angemessen reagieren zu können.
- System-Sicht: Der Programmierer muss nur die Bewegung bezüglich des Balls beschreiben. Die Umsetzung erfolgt implizit.

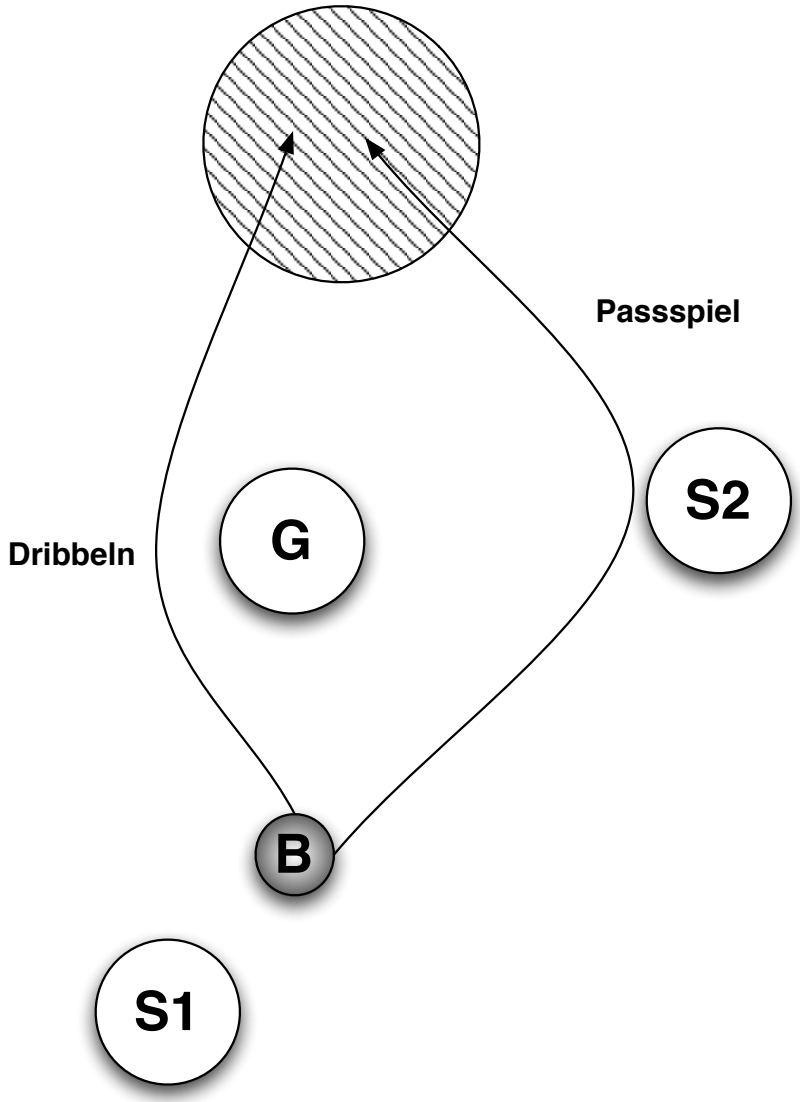


Abbildung: Anwendung Roboterfußball

Herausforderung

Alle genannten Anwendungen werden in **realer Raum-Zeit** ausgeführt:

- Es müssen zeitliche und räumliche Bedingungen berücksichtigt werden.
- Der Begriff der realen Raum-Zeit meint die Erweiterung des gängigen Echtzeitbegriffs um räumliche Dimensionen.

Problem: Wie kann ein Programmierer Anwendungen in realer Raum-Zeit entsprechend einer System-Sicht beschreiben?

Anforderungen an eine Programmierabstraktion

*A*1: **Reale Raum-Zeit:** Formulierung von räumlichen, zeitlichen und bewegungsspezifischen Eigenschaften und Beziehungen von bzw. zwischen Objekten

*A*2: **System-Sicht:** Beschreibung von Anwendungen für ein System, von dessen Verteiltheit abstrahiert wird

*A*3: Einfache intuitive Handhabung

Anforderungen an eine Programmierabstraktion

*A*1: **Reale Raum-Zeit:** Formulierung von räumlichen, zeitlichen und bewegungsspezifischen Eigenschaften und Beziehungen von bzw. zwischen Objekten

*A*2: **System-Sicht:** Beschreibung von Anwendungen für ein System, von dessen Verteiltheit abstrahiert wird

*A*3: Einfache intuitive Handhabung

$\Longrightarrow$ **Ansatz:** Erweiterung des konventionellen Anwendungscodes durch so genannte **Raum-Zeit-Constraints** (RZCs)

Raum-Zeit-Constraints

- Constraint (engl.): (Neben-) Bedingung, Beschränkung, Einschränkung
- RZCs beschreiben Einschränkungen in Raum, Zeit oder Bewegungen (Beispiel: 4-dimensionale Hüllkurve).
- Rein zeitliche RZCs sind bereits aus Echtzeitsystemen bekannt:
  - Periodendauer
  - Sollzeitpunkte
  - Präzedenzgraphen
  - ...
- RZCs rücken die Programmierung *der Anwendung* in den Vordergrund, anstatt der Programmierung des mobilen verteilten Systems.

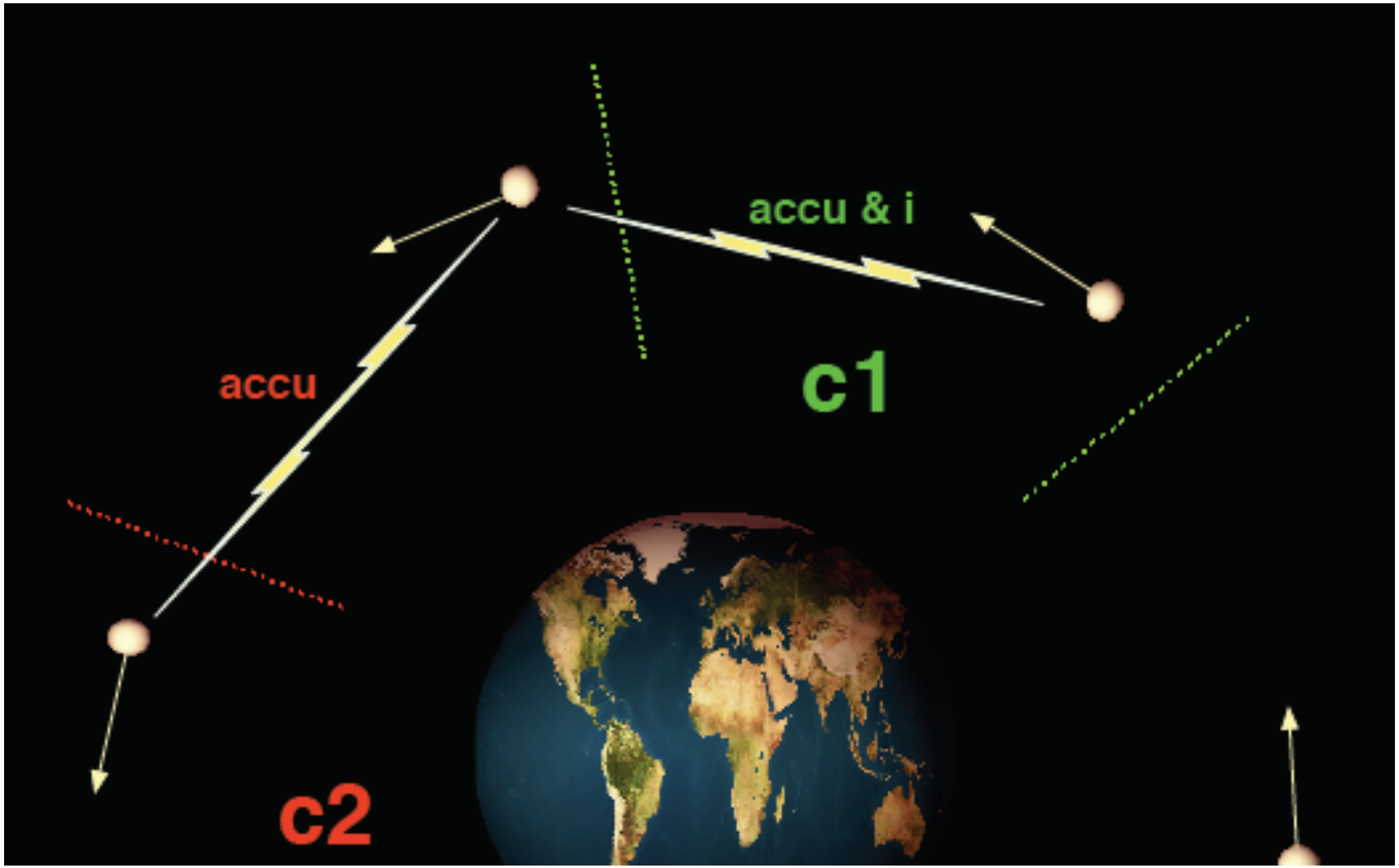
Herkömmlicher Anwendungscode:

```
int i;
int send_to_earth(int accu);

int main() {
  static int accu;
  accu = 0;

  while (true) {
    accu = accu + read_sensor();
    i = (i++)%1000;

    if (i == 0) send_to_earth(accu);

    sleep(1);
  }
  return 0;
}
```

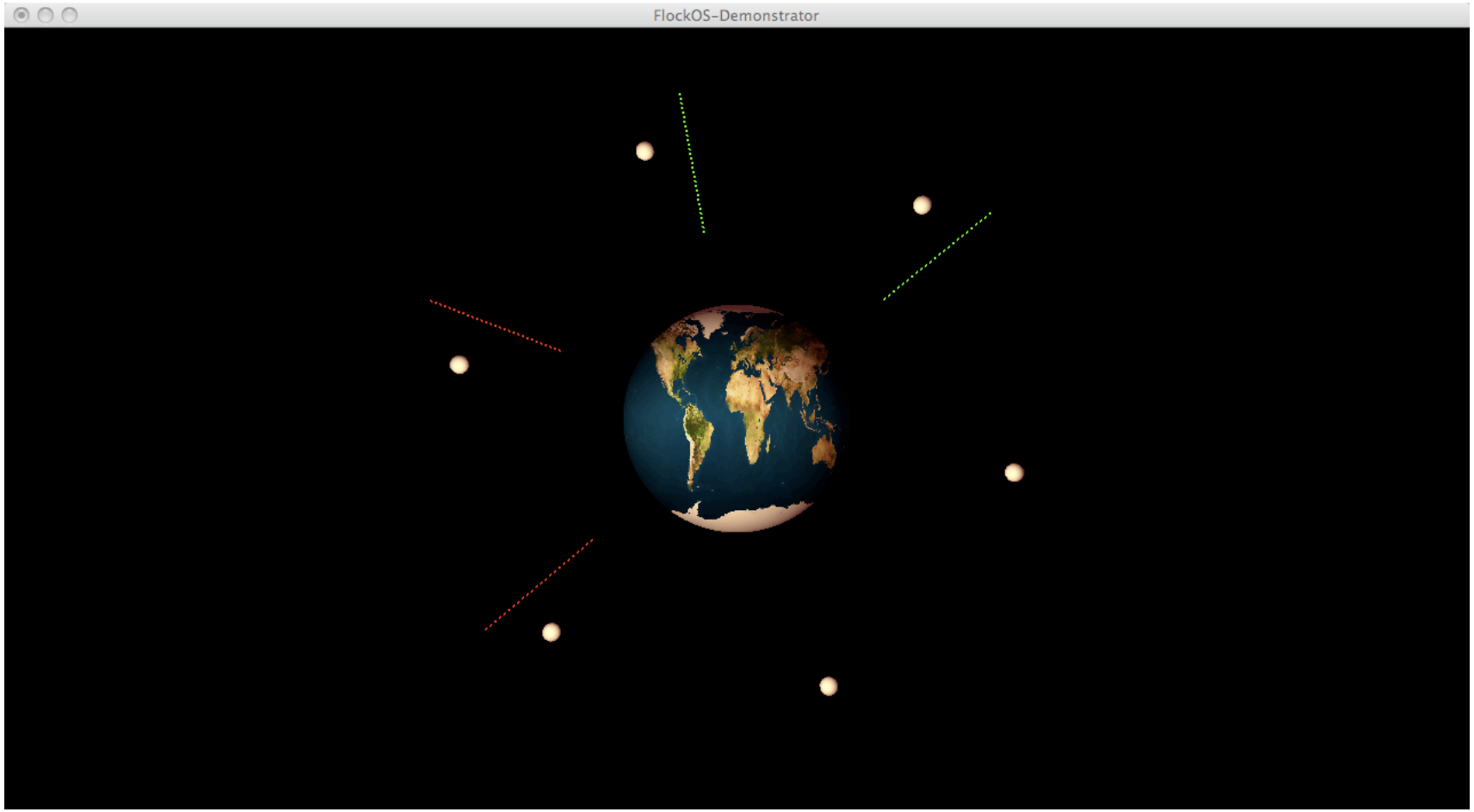
Annotierter Anwendungscode: (Auszug)

```
constraint_pol c1(phi,psi,r,t) = { ((int(phi) % 360)> CONST_A) &&
                                   ((int(phi) % 360) < CONST_B) };

constraint_pol c2(phi,psi,r,t) = { ((int(phi) % 360)> CONST_C) &&
                                   ((int(phi) % 360) < CONST_D) };

int i@c1;
int send_to_earth(int accu)@c2;

int main() {
  static int accu@c1;
  accu = 0;

  while (true) {
    accu = accu + read_sensor();
    i = (i++)%1000;

    if (i == 0) send_to_earth(accu);

    sleep(1);
  }
  return 0;
}
```

System-Sicht



Demonstration



Zusammenfassung

Einführung von **Raum-Zeit-Constraints** als eine Programmierabstraktion für Anwendungen in mobilen verteilten Systemen, die ...

- ... es erlaubt, Anwendungen an räumliche & zeitliche Sachverhalte zu knüpfen.
- ... von der Verteilung des ausführenden Systems abstrahiert und somit für den Programmierer transparent macht.
- ... eine Erweiterung konventionellen Codes darstellt.

Zusammenfassung

Einführung von **Raum-Zeit-Constraints** als eine Programmierabstraktion für Anwendungen in mobilen verteilten Systemen, die ...

- ... es erlaubt, Anwendungen an räumliche & zeitliche Sachverhalte zu knüpfen. $\Longrightarrow A1$ ✓
- ... von der Verteilung des ausführenden Systems abstrahiert und somit für den Programmierer transparent macht.
- ... eine Erweiterung konventionellen Codes darstellt.

Zusammenfassung

Einführung von **Raum-Zeit-Constraints** als eine Programmierabstraktion für Anwendungen in mobilen verteilten Systemen, die ...

- ...es erlaubt, Anwendungen an räumliche & zeitliche Sachverhalte zu knüpfen. $\Longrightarrow$ *A1* ✓
- ...von der Verteilung des ausführenden Systems abstrahiert und somit für den Programmierer transparent macht. $\Longrightarrow$ *A2* ✓
- ...eine Erweiterung konventionellen Codes darstellt.

Zusammenfassung

Einführung von **Raum-Zeit-Constraints** als eine Programmierabstraktion für Anwendungen in mobilen verteilten Systemen, die ...

- ...es erlaubt, Anwendungen an räumliche & zeitliche Sachverhalte zu knüpfen. $\Longrightarrow$ *A1* ✓
- ...von der Verteilung des ausführenden Systems abstrahiert und somit für den Programmierer transparent macht. $\Longrightarrow$ *A2* ✓
- ...eine Erweiterung konventionellen Codes darstellt. $\Longrightarrow$ *A3* ✓

Ausblick

- Formalisierung räumlicher, zeitlicher und bewegungsspezifischer Sachverhalte in einem (geschlossenen) Modell
- Festlegung der genauen Syntax der Raum-Zeit-Constraints
- Integration der Erweiterungen in gängige Programmiersprachen
- Entwickeln von generischen Methoden zum Finden eines Ablaufplanes für komplexere Anwendungen

Ende des Vortrags

Vielen Dank für Ihre Aufmerksamkeit.
Fragen?

Codegenerierung

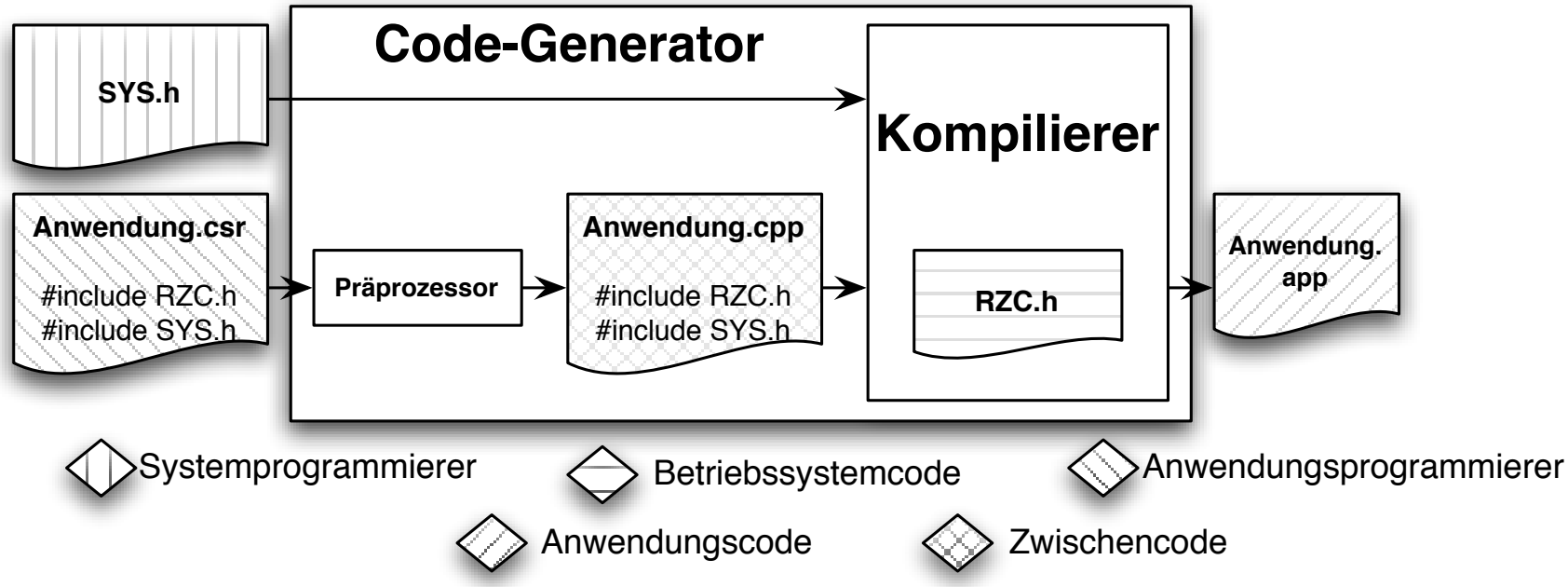


Abbildung: Generierung des verteilten Anwendungscodes

Laufzeitumgebung

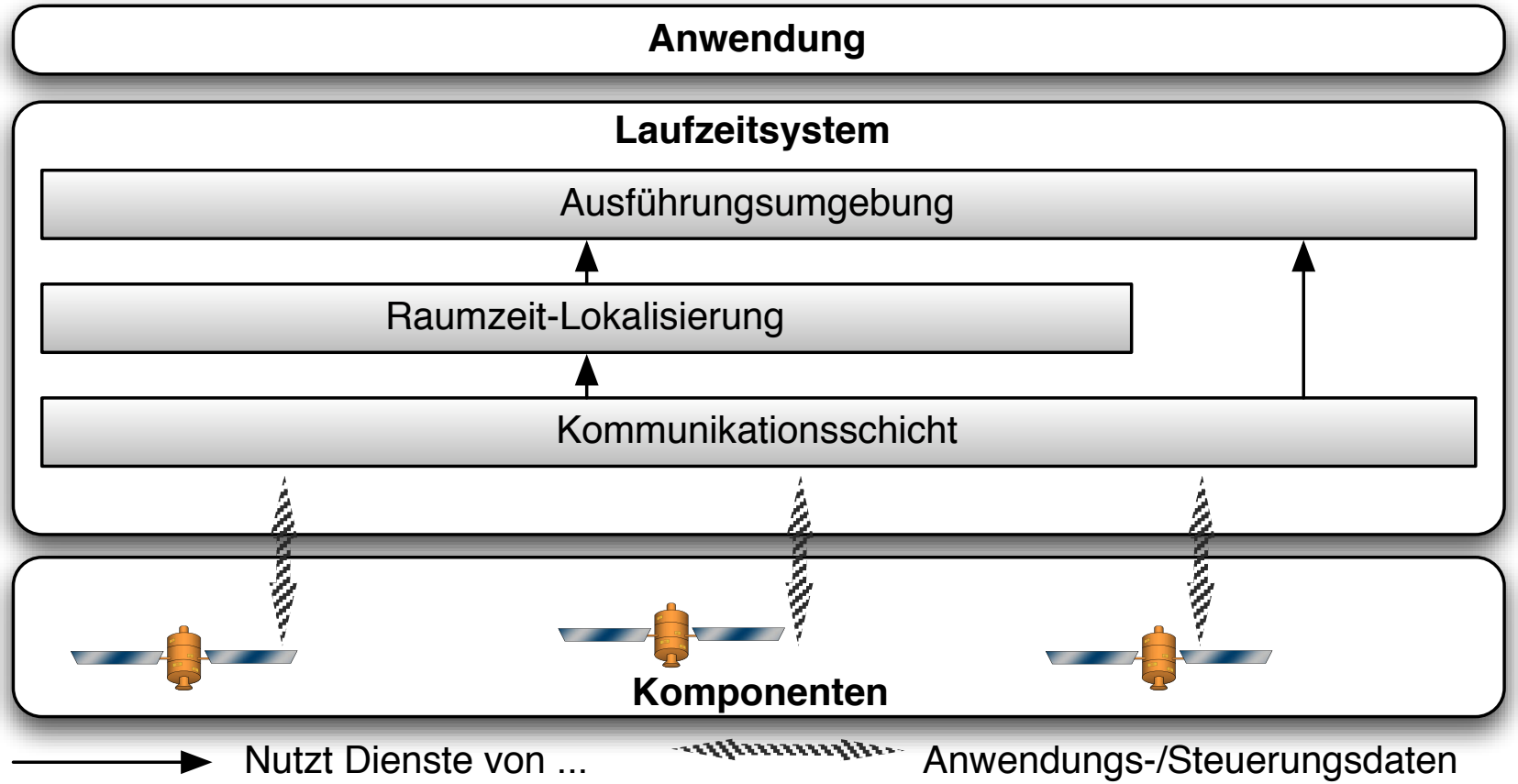


Abbildung: Laufzeitumgebung von FlockOS