

Proseminar Technische Informatik

The Byzantine Generals' Problem

Esra Ünal

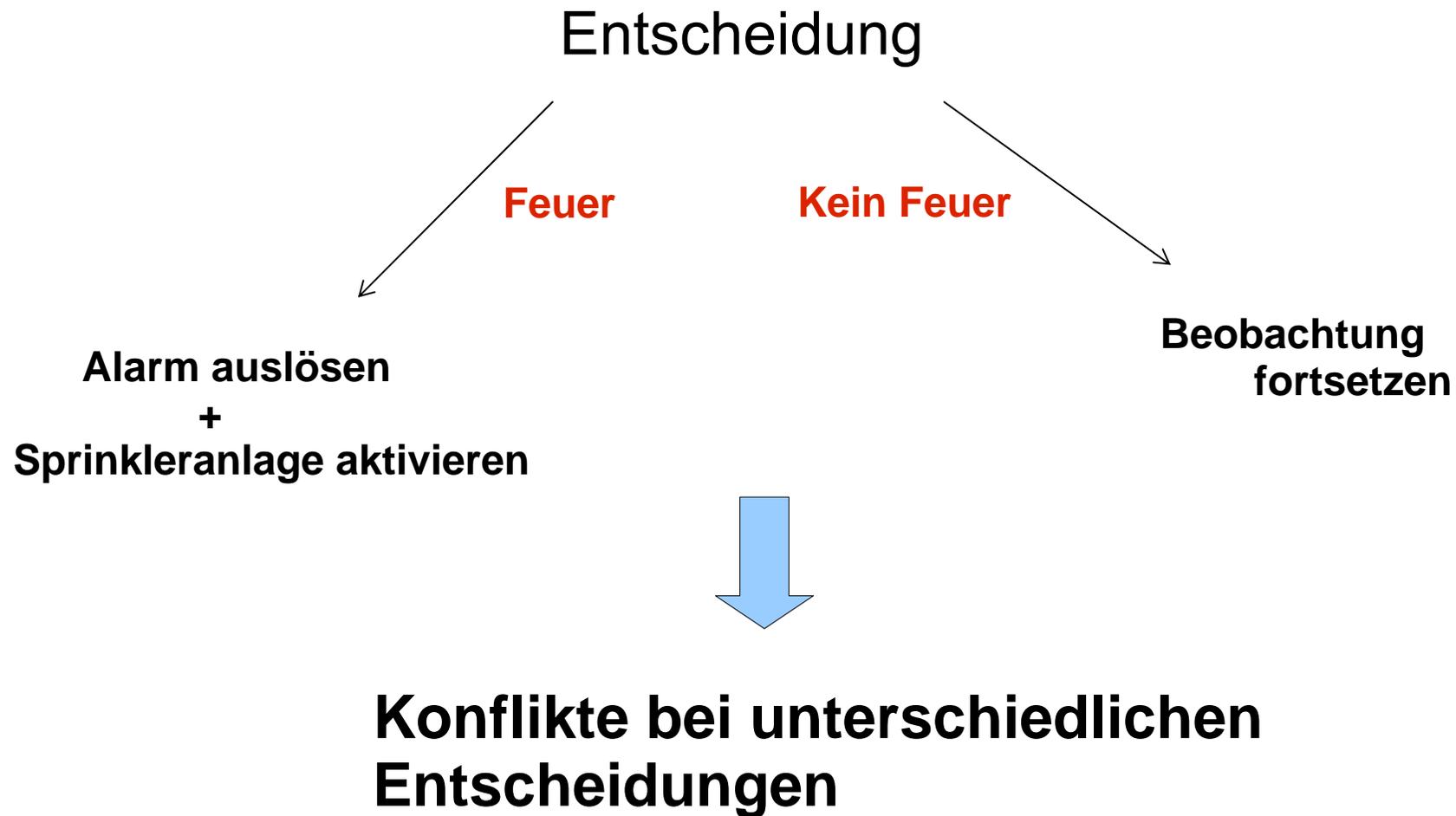
Gliederung

- 1. Beispiel: Feuermeldeanlage**
 - 2. Formalisierung des Problems**
 - 3. Definition**
 - 4. Ursprung der Namensgebung**
 - 5. Voraussetzungen für die Lösung**
 - 6. Lösung**
 - 7. Erweiterung**
 - 8. Zusammenfassung**
 - 9. Quellen**
-

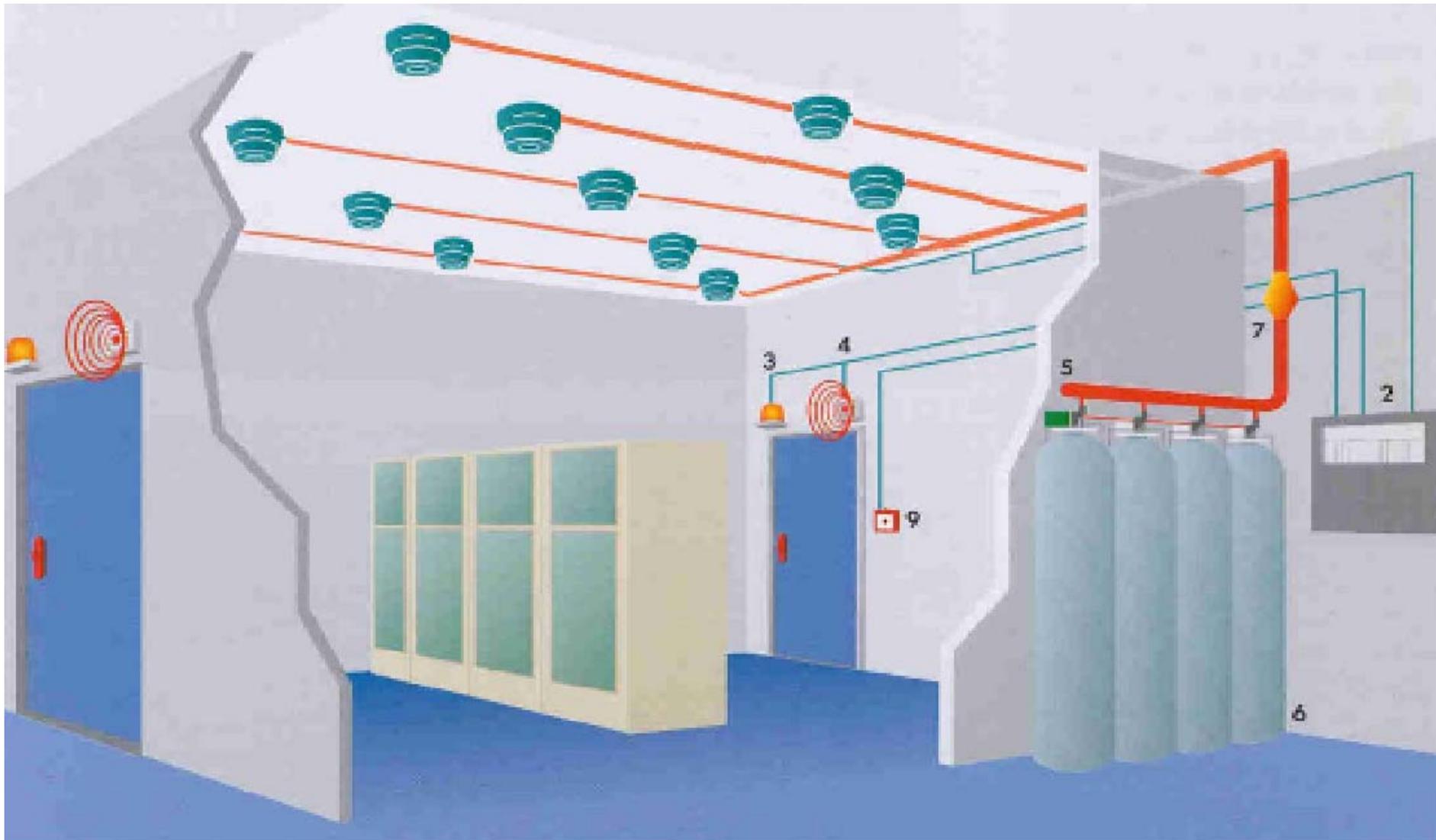
Beispiel: Feuermeldeanlage – Beschreibung(1)

- Besteht aus einem Netzwerk von Temperatursensoren
- alle Sensoren können mit den anderen kommunizieren
- Entscheidung wird mithilfe der Nachrichten von den anderen Sensoren getroffen
- Jeder Sensoren entscheidet für sich, ob es seinen Teil der Löschanlage startet oder nicht

Beispiel: Feuermeldeanlage – Beschreibung(2)



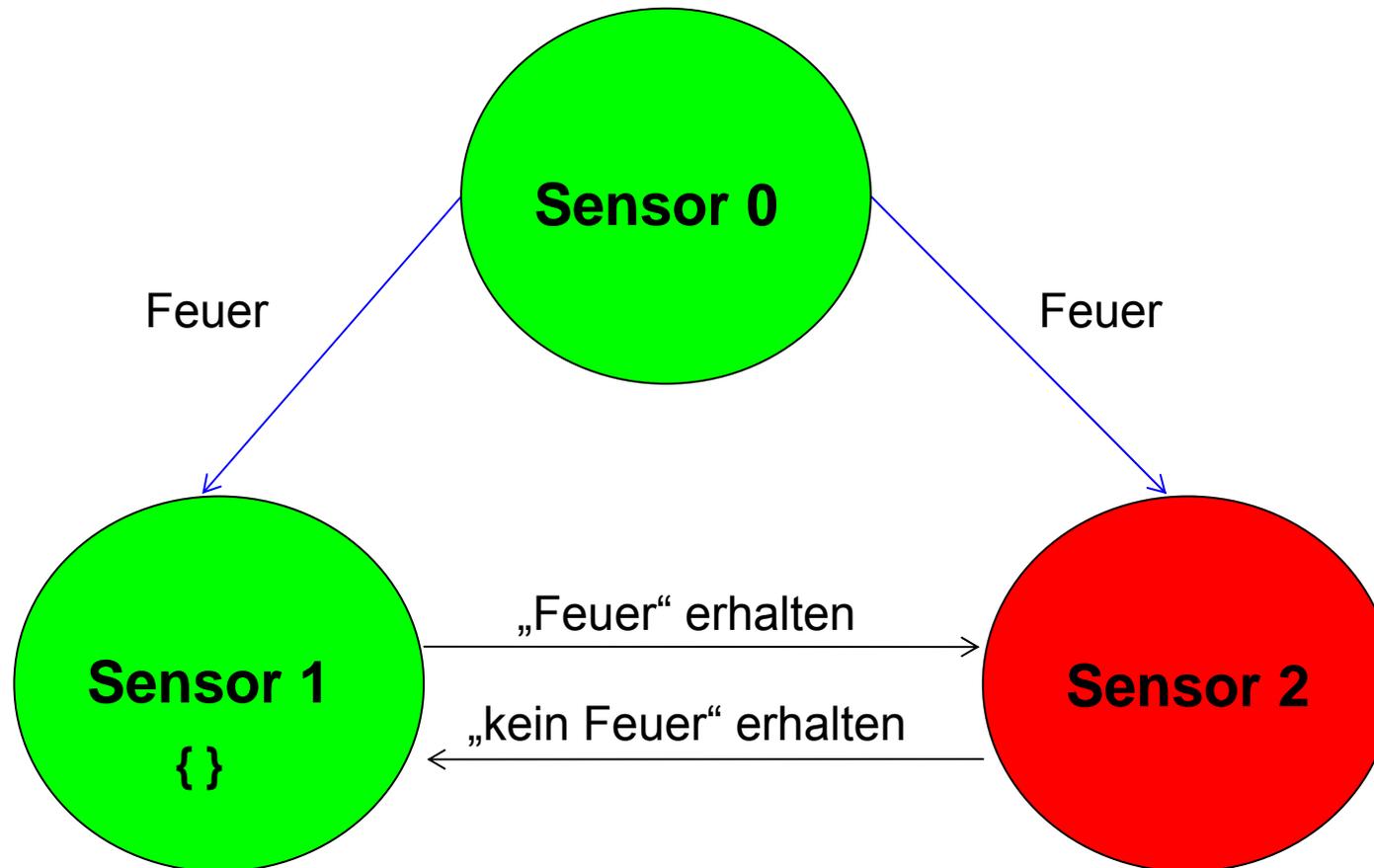
Beispiel: Feuermeldeanlage- Beschreibung(3)



Beispiel: Feuermeldeanlage- Beschreibung(4)

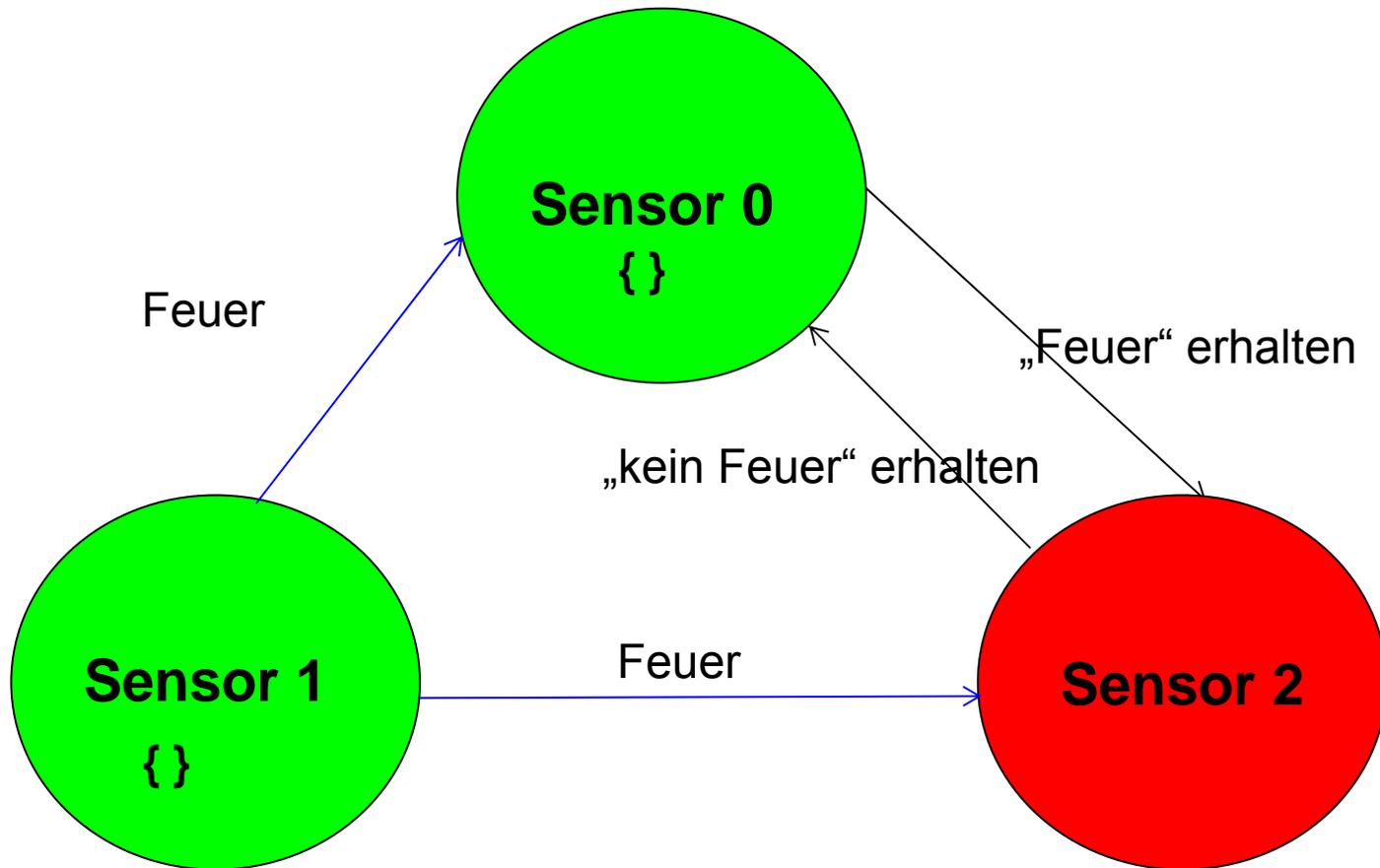


Beispiel: Feuermeldeanlage – Fall mit 3 Sensoren(1)



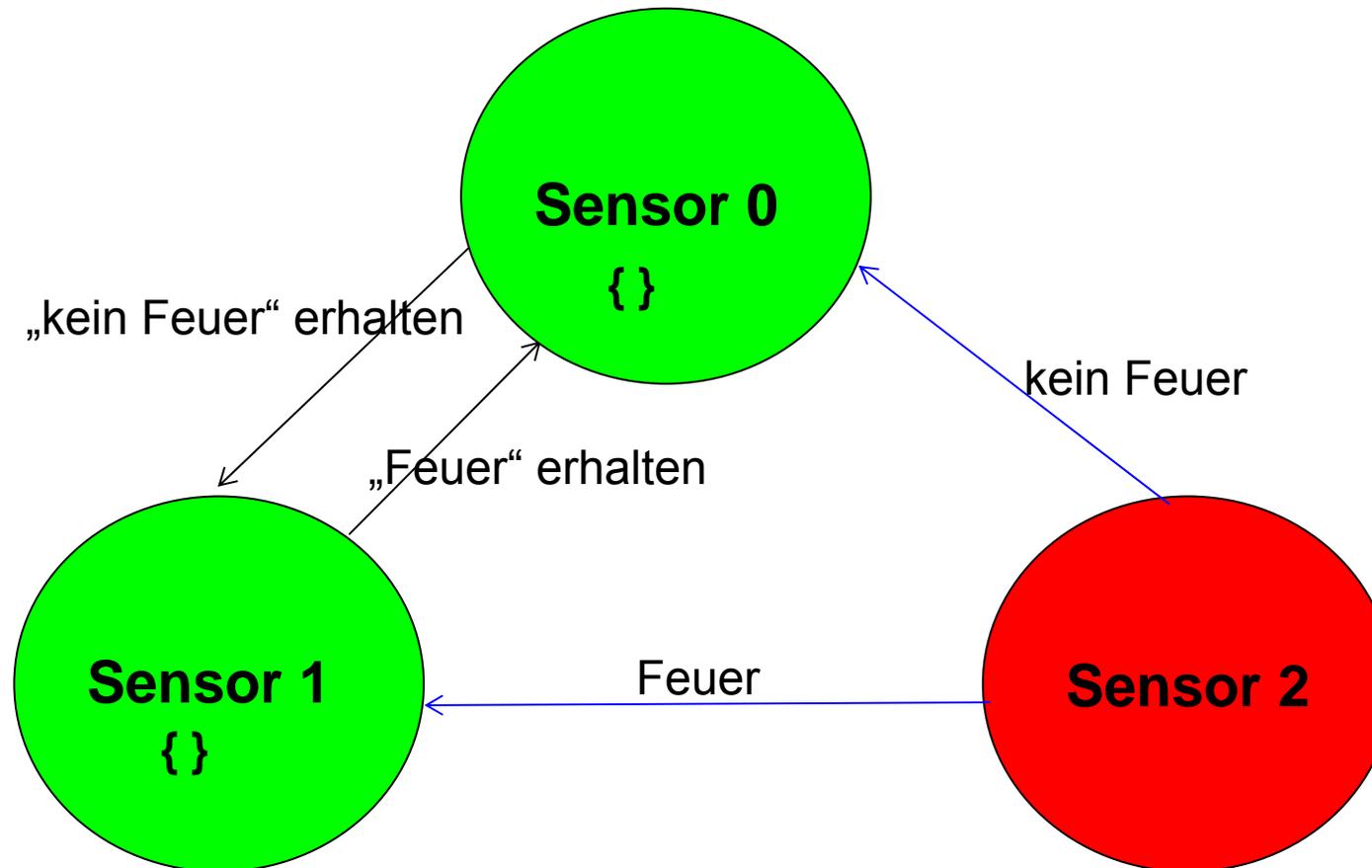
Server 0 sendet sein Ergebnis an alle

Beispiel: Feuermeldeanlage – Fall mit 3 Sensoren(2)



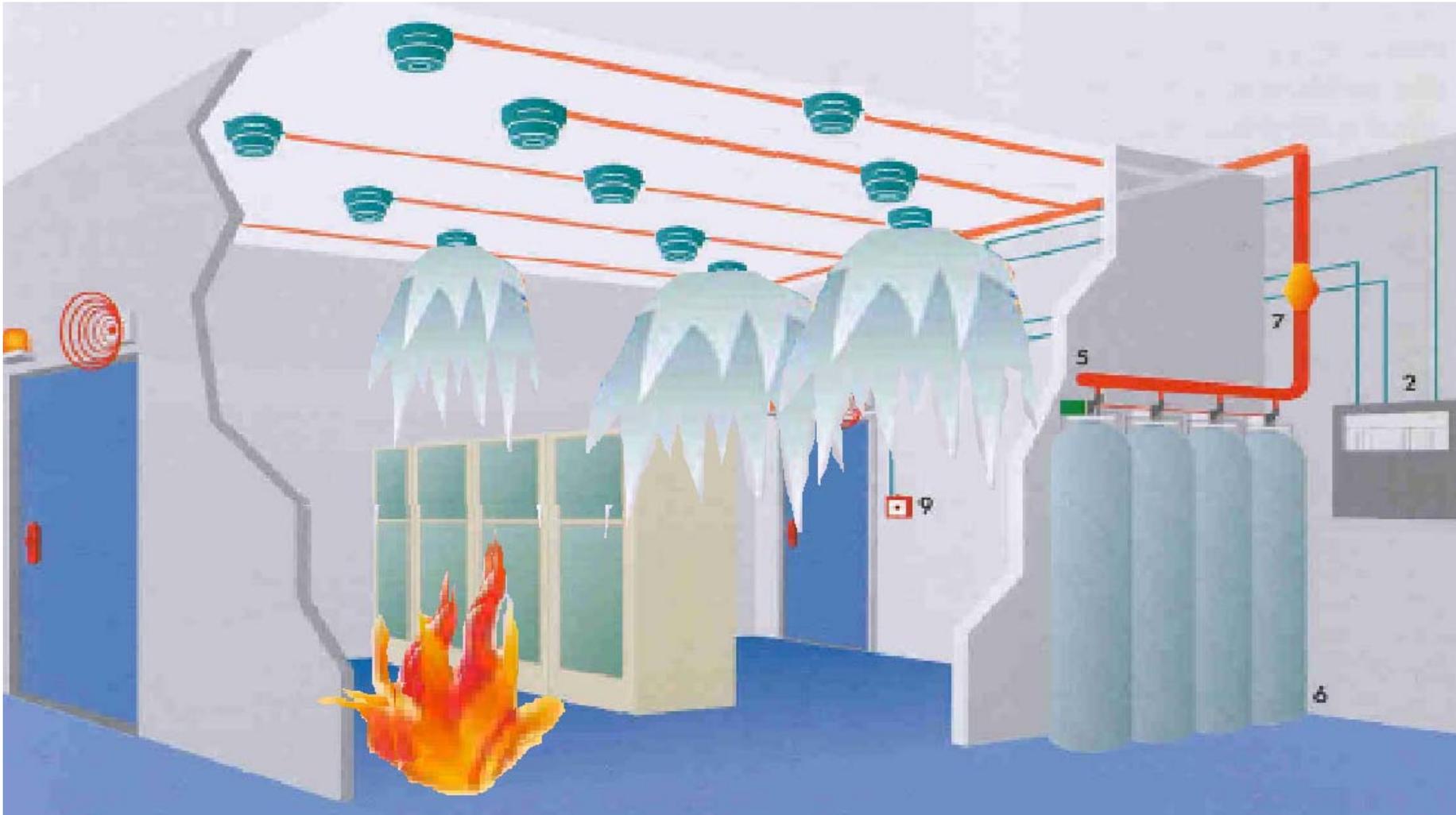
Server 1 sendet sein Ergebnis an alle

Beispiel: Feuermeldeanlage – Fall mit 3 Sensoren(3)



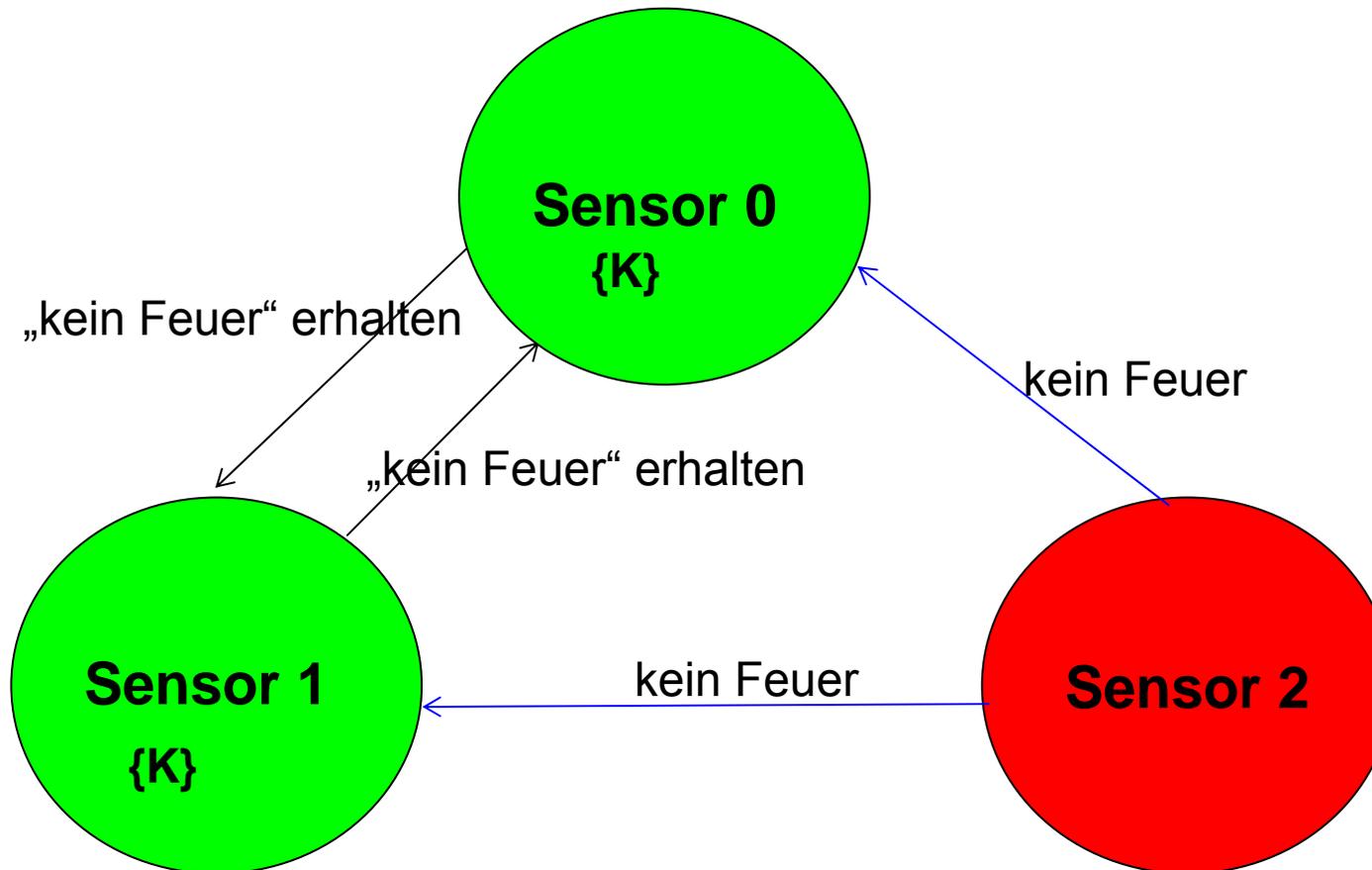
Server 2 sendet sein Ergebnis an alle

Beispiel: Feuermeldeanlage – Fall mit 3 Sensoren(4)



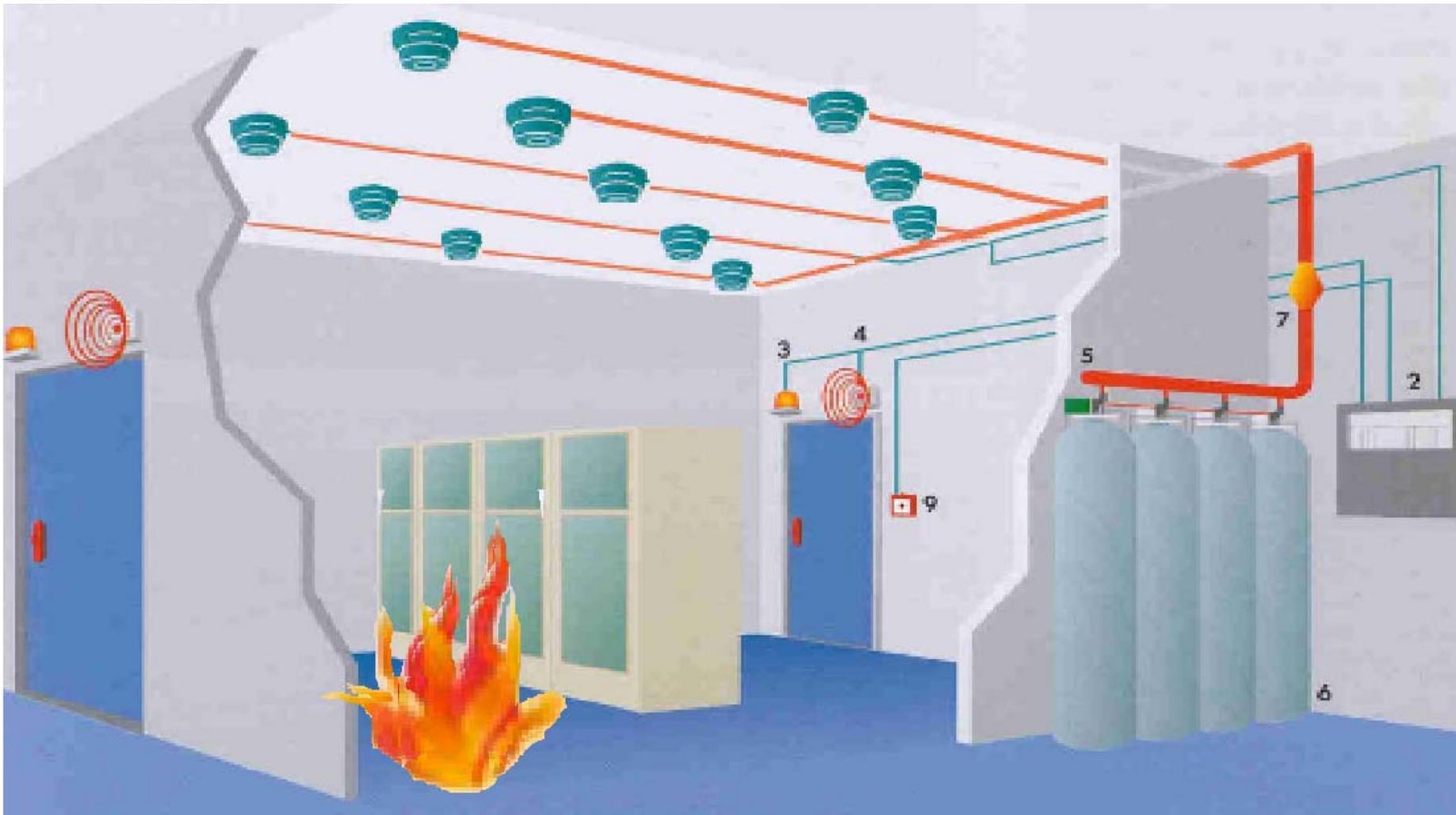
Nur einige Löschanlagen sind aktiv, da die Ergebnismengen leer sind (undefinierter Zustand)

Beispiel: Feuermeldeanlage – Fall mit 3 Sensoren(3)



Server 2 sendet sein Ergebnis an alle

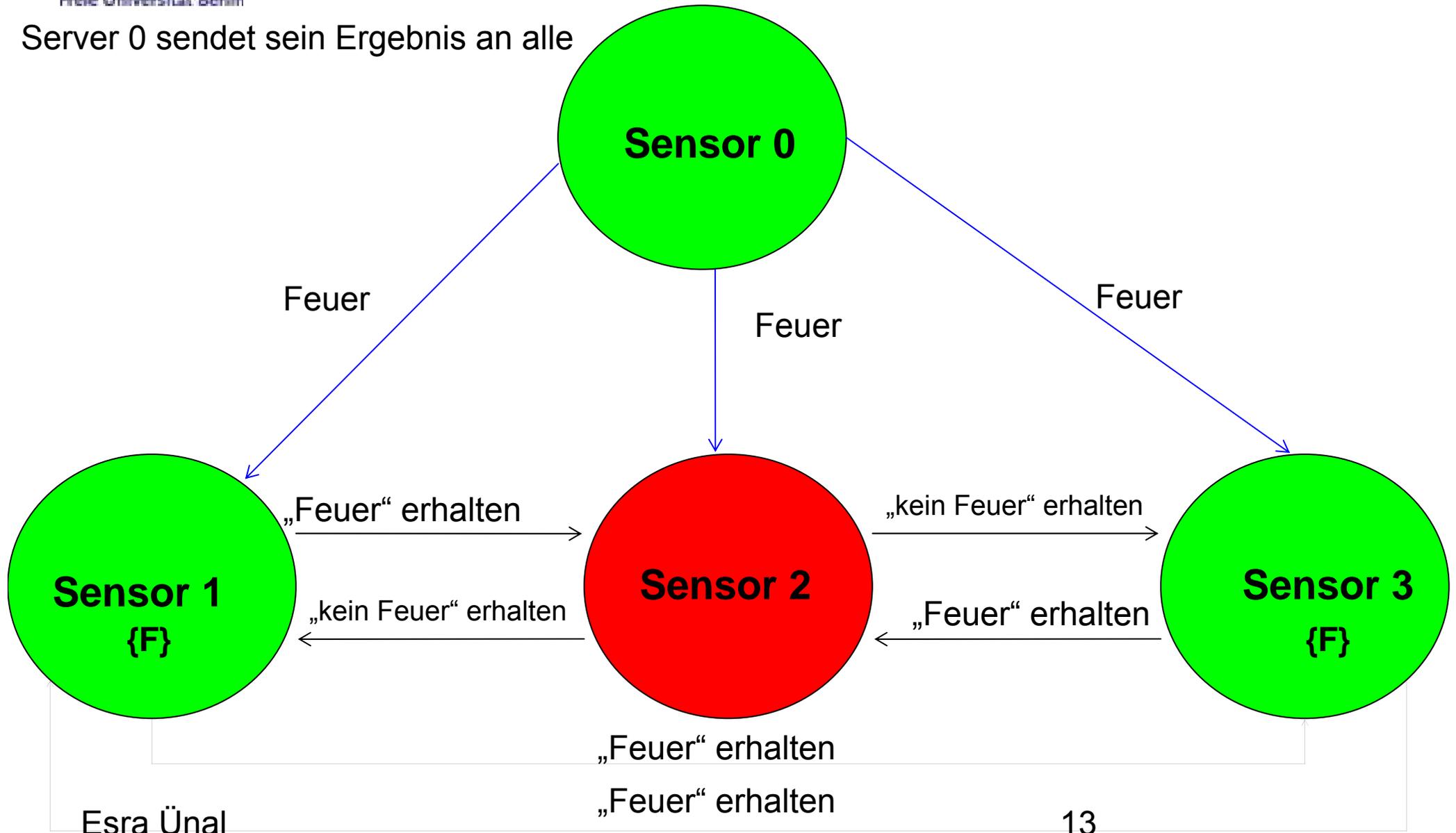
Beispiel: Feuermeldeanlage – Fall mit 3 Sensoren(6)



Löschanlagen bleiben aus, weil die „Mehrheit“ der Sensoren dagegen ist

Beispiel: Feuermeldeanlage – Fall mit 4 Sensoren

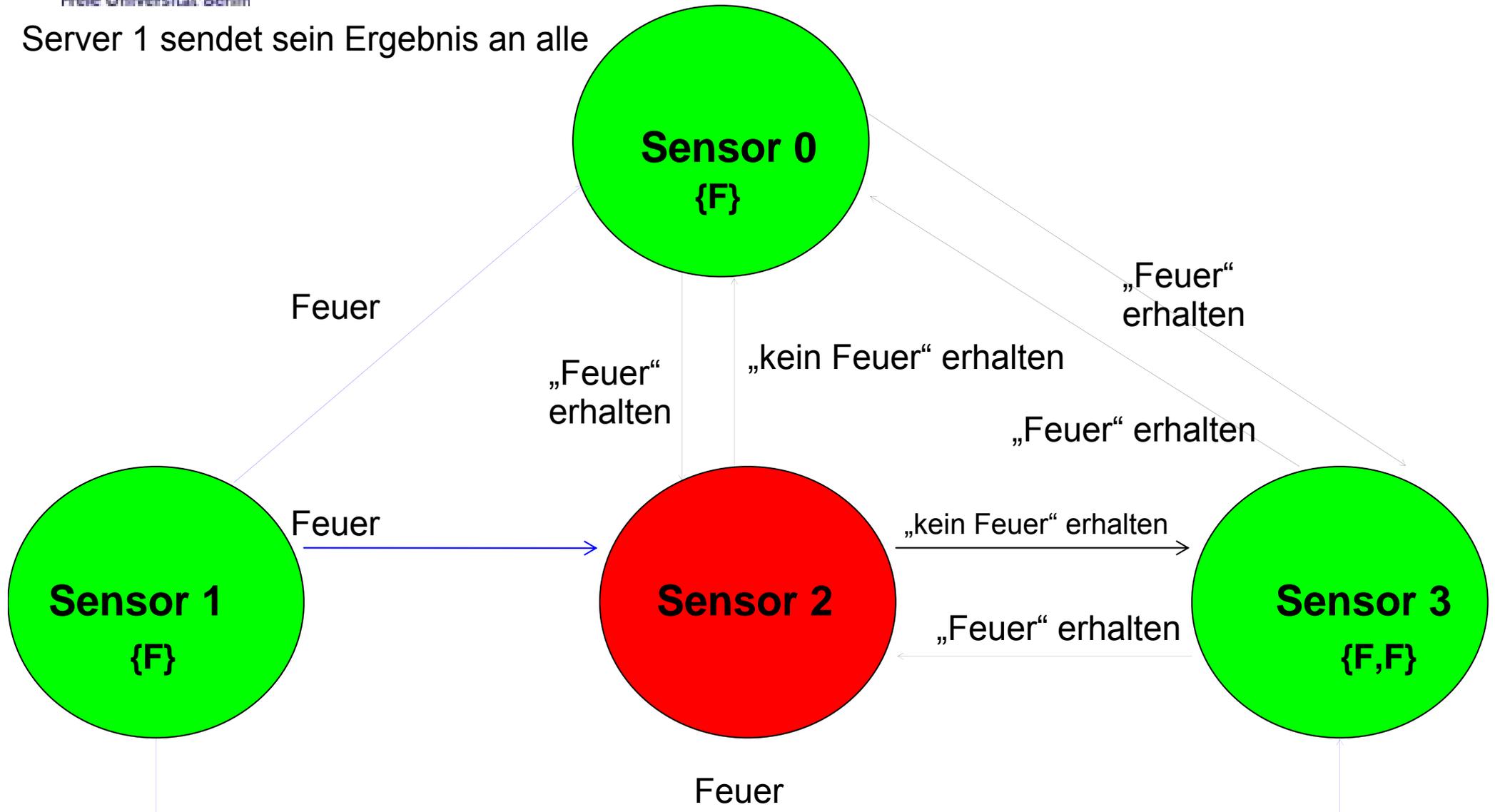
Server 0 sendet sein Ergebnis an alle



Beispiel: Feuermeldeanlage – Fall mit 4 Sensoren



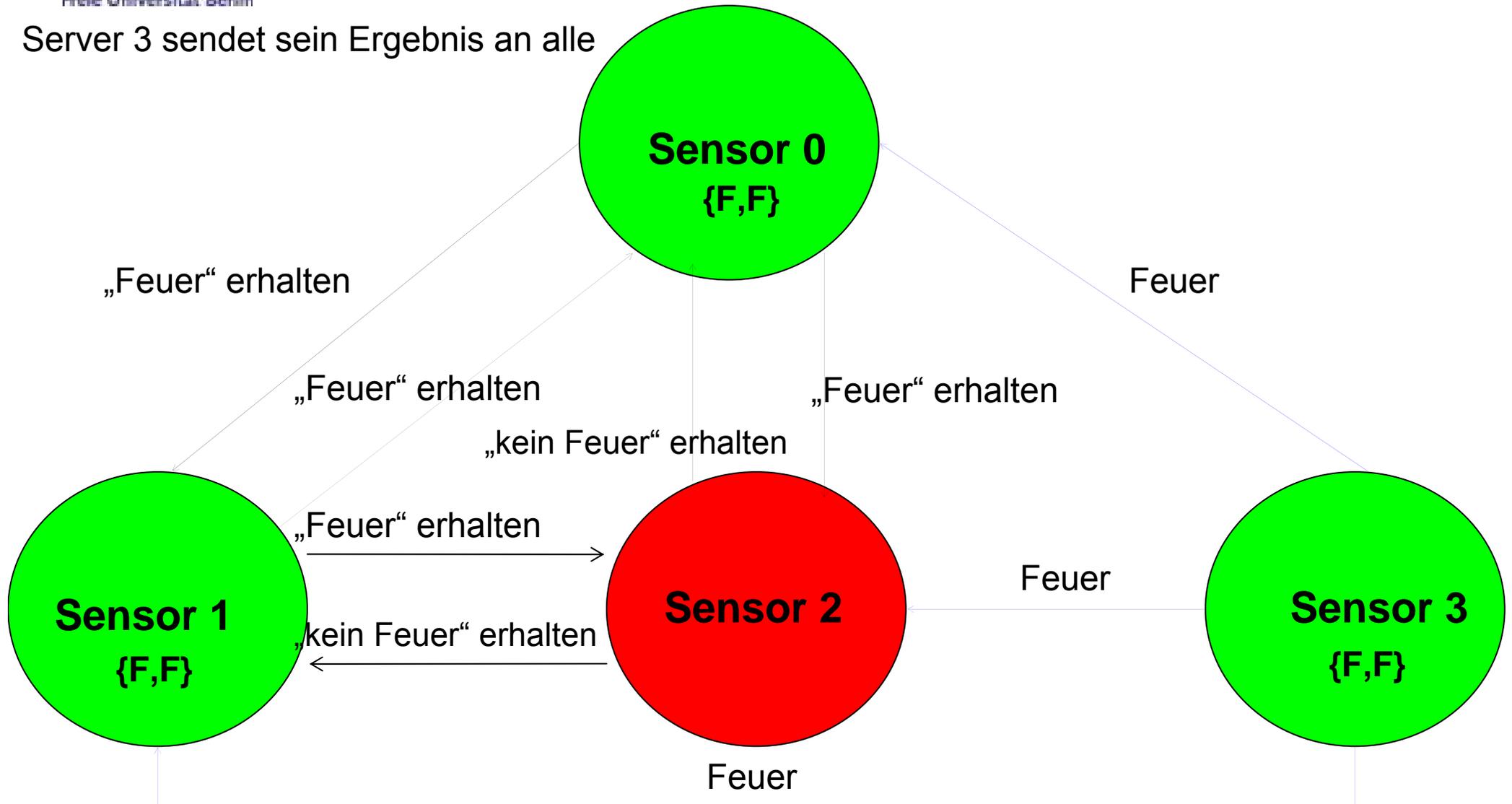
Server 1 sendet sein Ergebnis an alle



Beispiel: Feuermeldeanlage – Fall mit 4 Sensoren



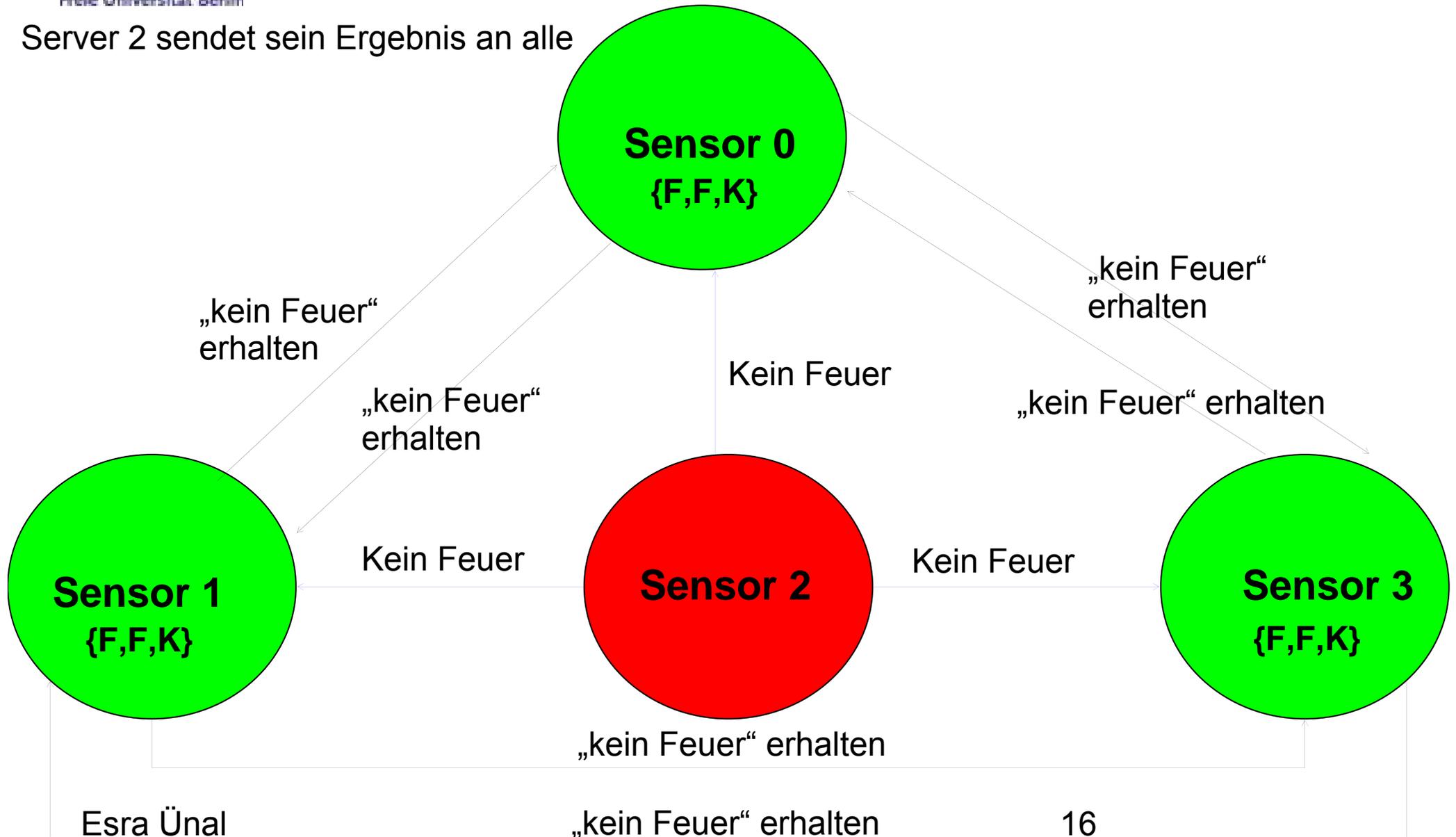
Server 3 sendet sein Ergebnis an alle



Beispiel: Feuermeldeanlage – Fall mit 4 Sensoren



Server 2 sendet sein Ergebnis an alle



Beispiel: Feuermeldeanlage – Fall mit 4 Sensoren



Alle funktionierenden Löschanlagen gehen an
Esra Ünal

Formalisierung des Problems

- System besteht aus n Komponenten
 - jede Komponente i versendet Nachricht v_i an alle anderen $n-1$ Komponenten
 - Master: Komponente, die die erste Nachricht versendet
 - Menge V_i speichert die Nachrichten der anderen Komponenten
 - Die Menge V_i dient als Entscheidungsgrundlage für künftige Aktionen
- ➡ Ziel: alle korrekten Komponenten treffen richtige Entscheidung

Definition

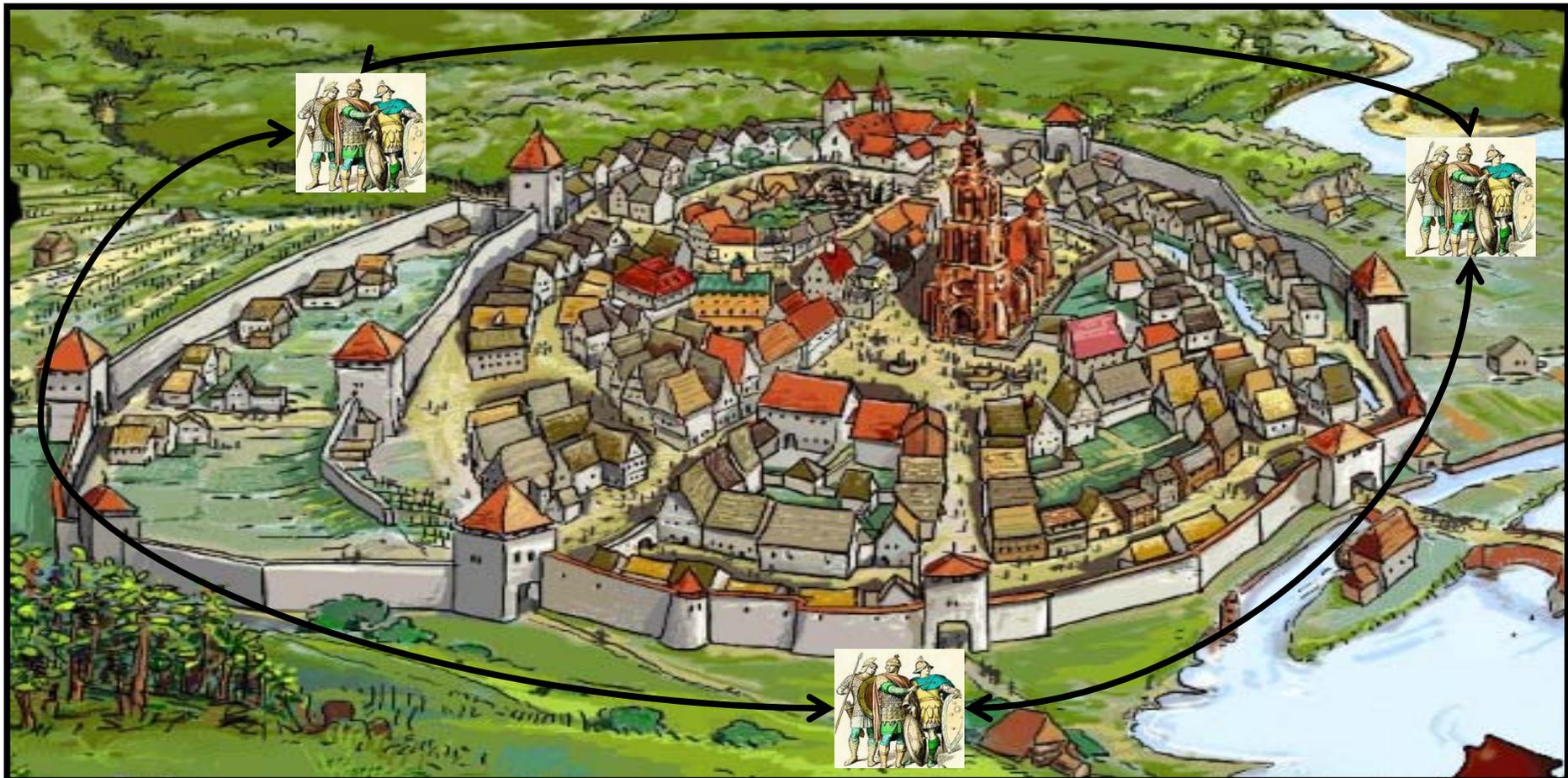
**A reliable computer system must be able to cope with the failure of one or more of its components.
A failed component may exhibit a type of behavior that is often overlooked - namely, sending conflicting information to different parts of the system.**

The problem of coping with this type of failure is expressed abstractly as the Byzantine Generals Problem.

Quelle: *The Byzantine Generals Problem*

Ursprung der Namensgebung

Modell: Das Problem der Byzantinischen Generäle



Voraussetzungen für die Lösung

1. min. **$3f+1$** -Komponenten nötig, für die Tolerierung von max. **f** fehlerhaften Bestandteilen
2. Alle funktionstüchtigen Sensoren treffen dieselbe Entscheidung
3. Falls der Mastersensor korrekt funktioniert, dann müssen alle anderen intakten Sensoren seinen Befehl befolgen

Lösung - Algorithmus

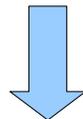
- i. Mastersensor sendet an alle Sensoren Ergebnis v_0
- ii. Jeder Sensor i nimmt diesen Wert in seine Menge V_i auf oder Default-Wert
- iii. Für jedes i gilt:
Sensor i sendet an alle $n-1$ Sensoren sein Wert v_i
- iv. Wähle den Wert, der in V_i am häufigsten auftaucht

Lösung - Problematik

- Algorithmus verlangt zwischen allen Sensoren direkte Kommunikationspfade
- Bei n Sensoren entspricht es genau

$$\frac{n(n-1)}{2} \text{ Pfade}$$

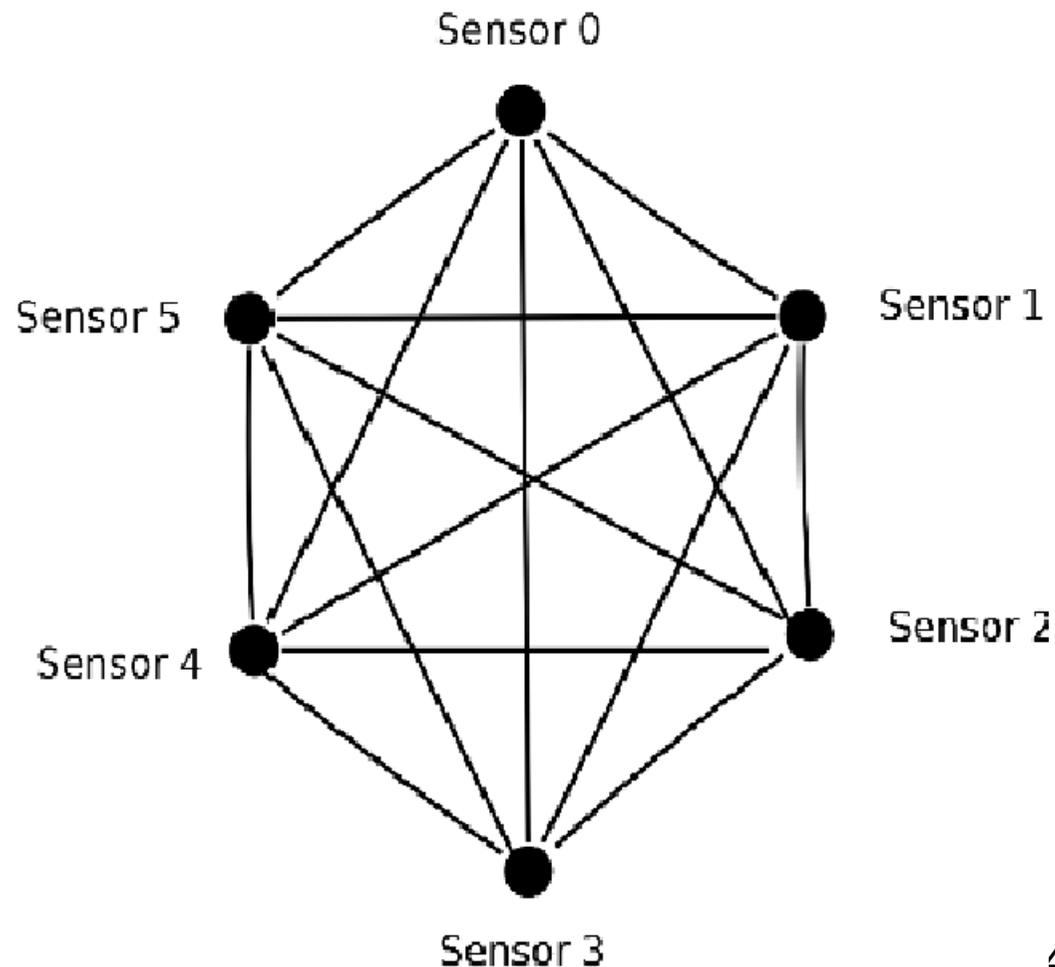
- Kommunikation verursacht auch Kosten



Erweiterungen erforderlich

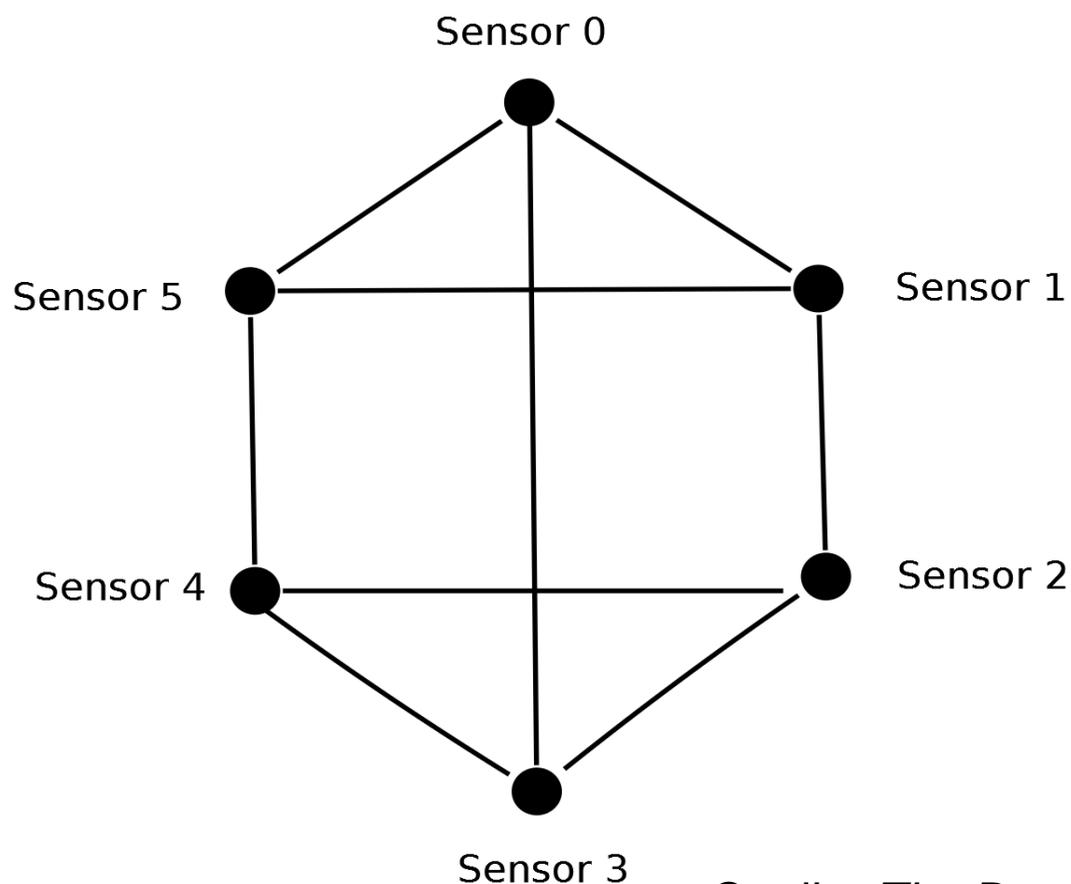
Erweiterung – Reduzierung der Kommunikationspfade(1)

Ein vollständiger Graph mit sechs Knoten:



Erweiterung – Reduzierung der Kommunikationspfade(2)

Ein 3-regulärer Graph:



Quelle: *The Byzantine Generals Problem*

Zusammenfassung (1)

- **Wo** tauchen Byzantinische Fehler auf?

in verteilten Systemen

- **Wann** tauchen sie auf?

nicht vorhersehbar

- **Warum** führen sie zu Fehlern?

die falsch berechneten Teilergebnisse sind von richtigen nicht unterscheidbar

- **Was** sind die Folgen?

unbemerkte Störung der gesamten Systemausführung

Zusammenfassung (2)

• **Voraussetzung für die Lösung:**

Für die Tolerierung von f Fehlern muss das System aus $3f+1$ Komponenten bestehen.

• **Lösungsidee:**

- › Alle Komponenten tauschen sich über ihre ermittelten Werte aus
- › Der häufigste Wert wird gewählt

• **Erweiterung:**

- › Reduzierung der Kommunikationspfade

Quellen

- LAMPORT, L. ; SHOSTAK, R. ; PEASE, M.: The Byzantine Generals Problem
Transactions on Programming Languages and Systems, 1982, S. 382–401
- DRISCOLL, Kevin ; HALL, Brendan ; SIVENCRONA, Hakan ; ZUMSTEG, Phil: Byzantine Fault Tolerance, From Theory To Reality. In: Computer Safety, Reliability, and Security Bd. 2788/2003. Berlin Heidelberg : Springer-Verlag, September 2003, S. 235–248

Bilder:

- http://www2.hs-esslingen.de/fachbereiche/vu/VU_aktuell/total.html
- http://www.thg.fr.bw.schule.de/silkeamberg/stadtbild/stadt_mittelalter_swr.jpg
- <http://www.kriegsreisende.de/antike/antik-img/byzanz.jpg>

Fragen