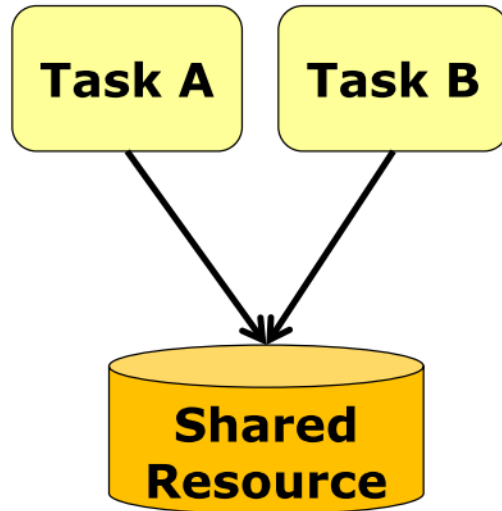


Recap Semaphore & Mutex

Manuel Schär, Robin Baumgartner

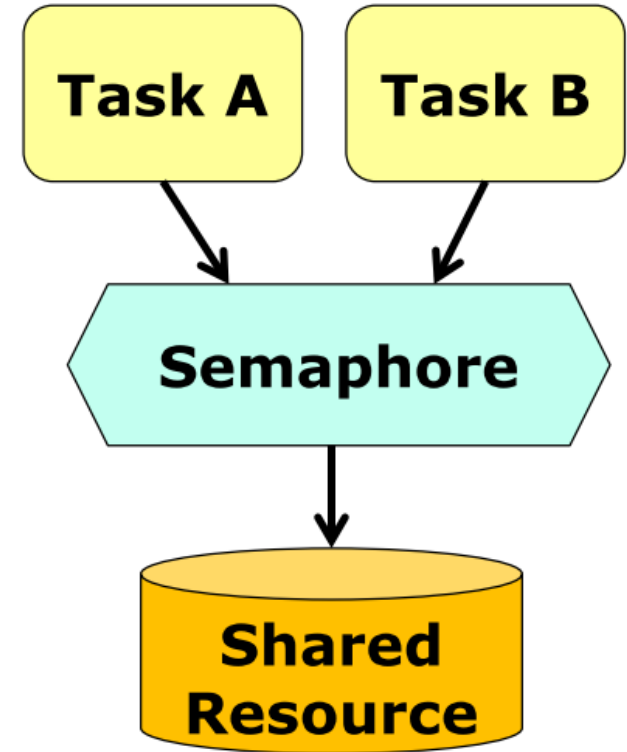
Problem: Gemeinsame Ressourcen

- Führt zu unvorhersehbarem Verhalten



Lösung: Semaphore

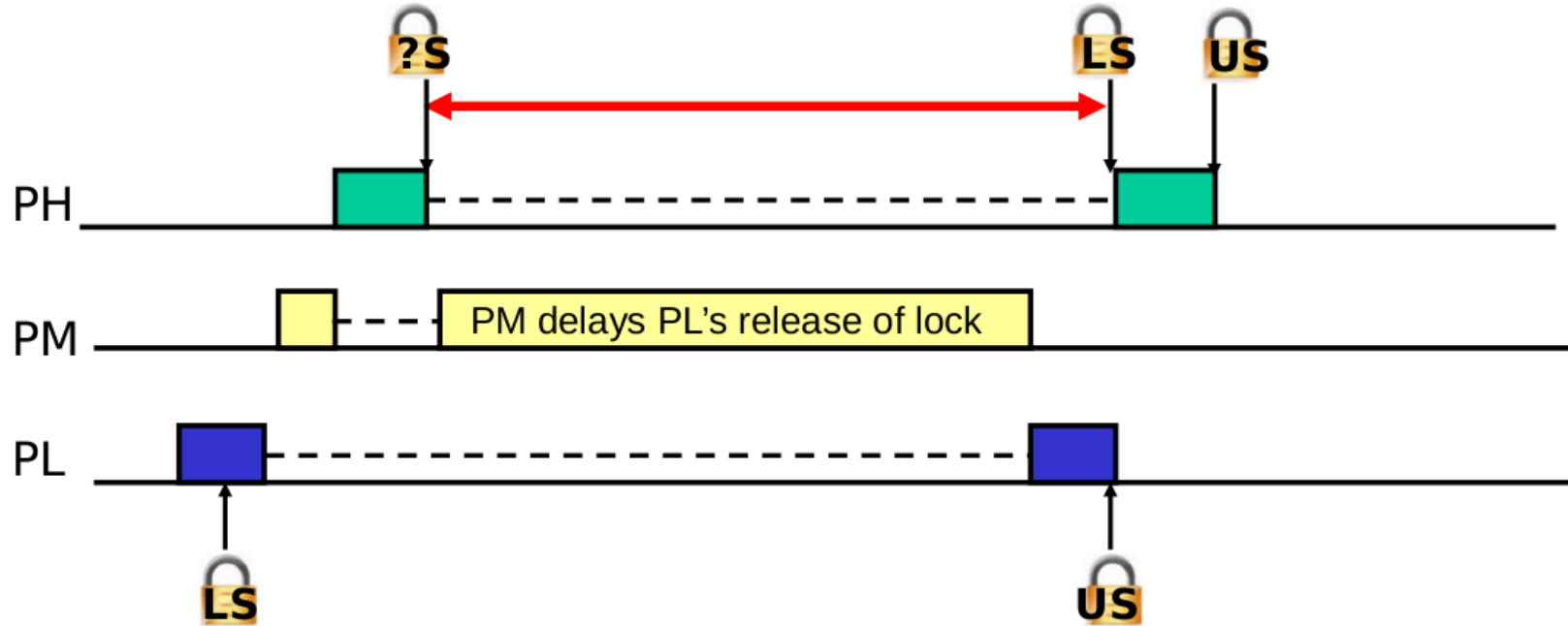
- Synchronisation mit Semaphore regelt den Zugriff auf die gemeinsame Ressource.



Synchronisation

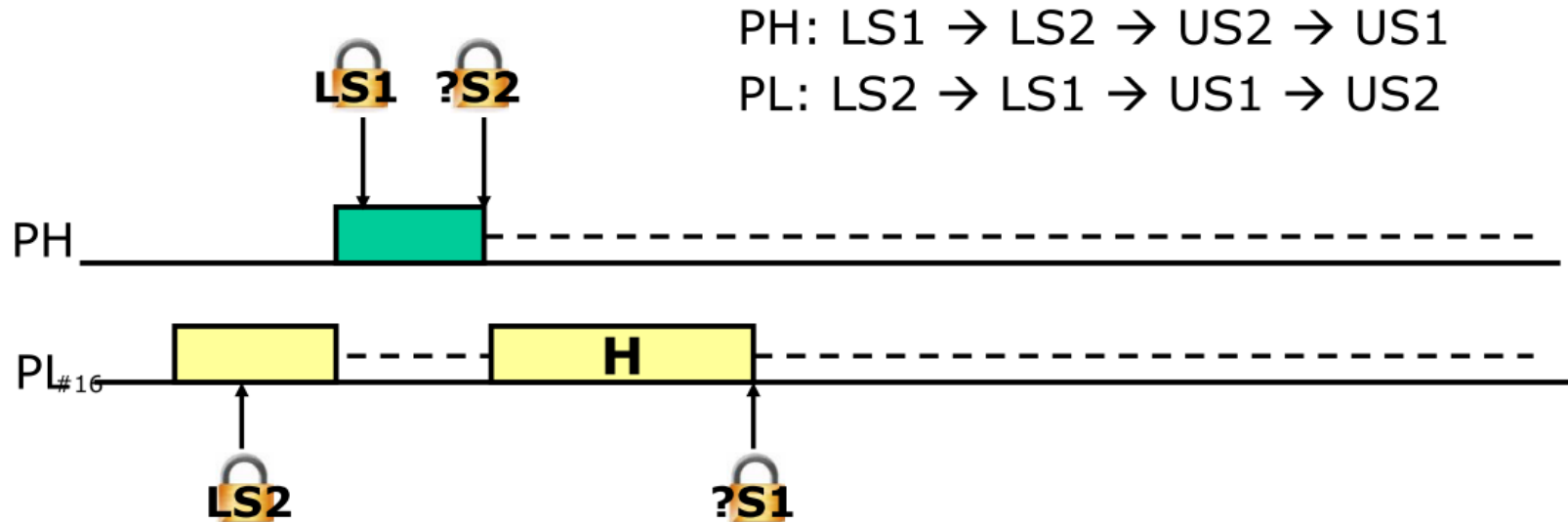
- Critical section
 - Muss unterbrechungsfrei ablaufen
- Semaphore
 - Synchronisationsmechanismus
- Mutual Exclusion (Mutex)
 - Spezialfall einer binären Semaphore

Problem: Priority Inversion



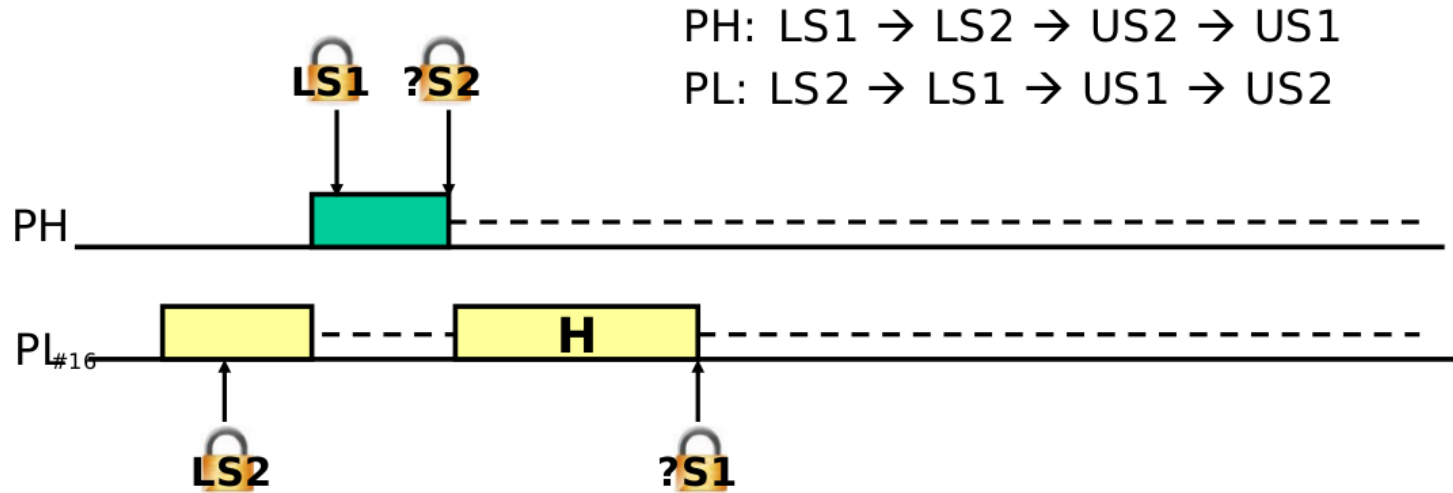
Lösung: Priority Inheritance

- Priorität der wartenden Tasks wird an den Task mit dem Lock vererbt



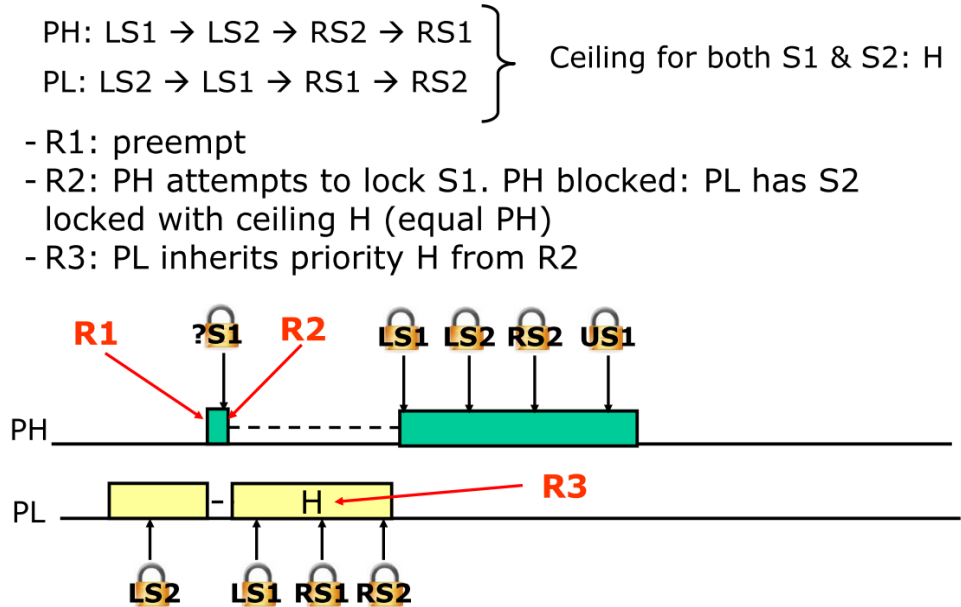
Problem: Deadlocks

- Priority Inheritance löst nur das Problem der Priority Inversion

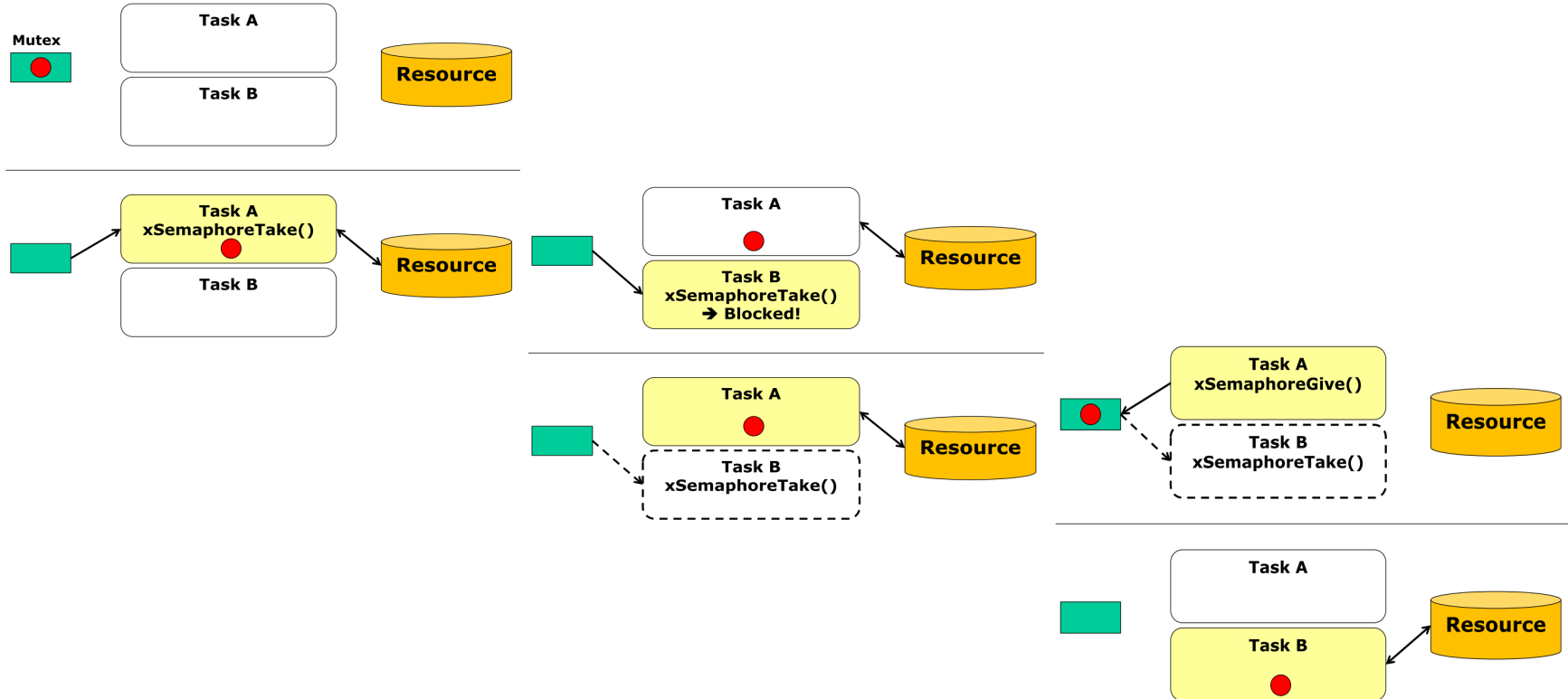


Lösung: Priority Ceiling

- Der Task bekommt die Ressource nur wenn seine Priorität höher als die aktuelle Prioritätsgrenze (Priority Ceiling) ist.



FreeRTOS Binary Semaphores



IPC mit Binary Semaphore

- Mit binären Semaphoren lassen sich Producer-Consumer-Systeme realisieren

