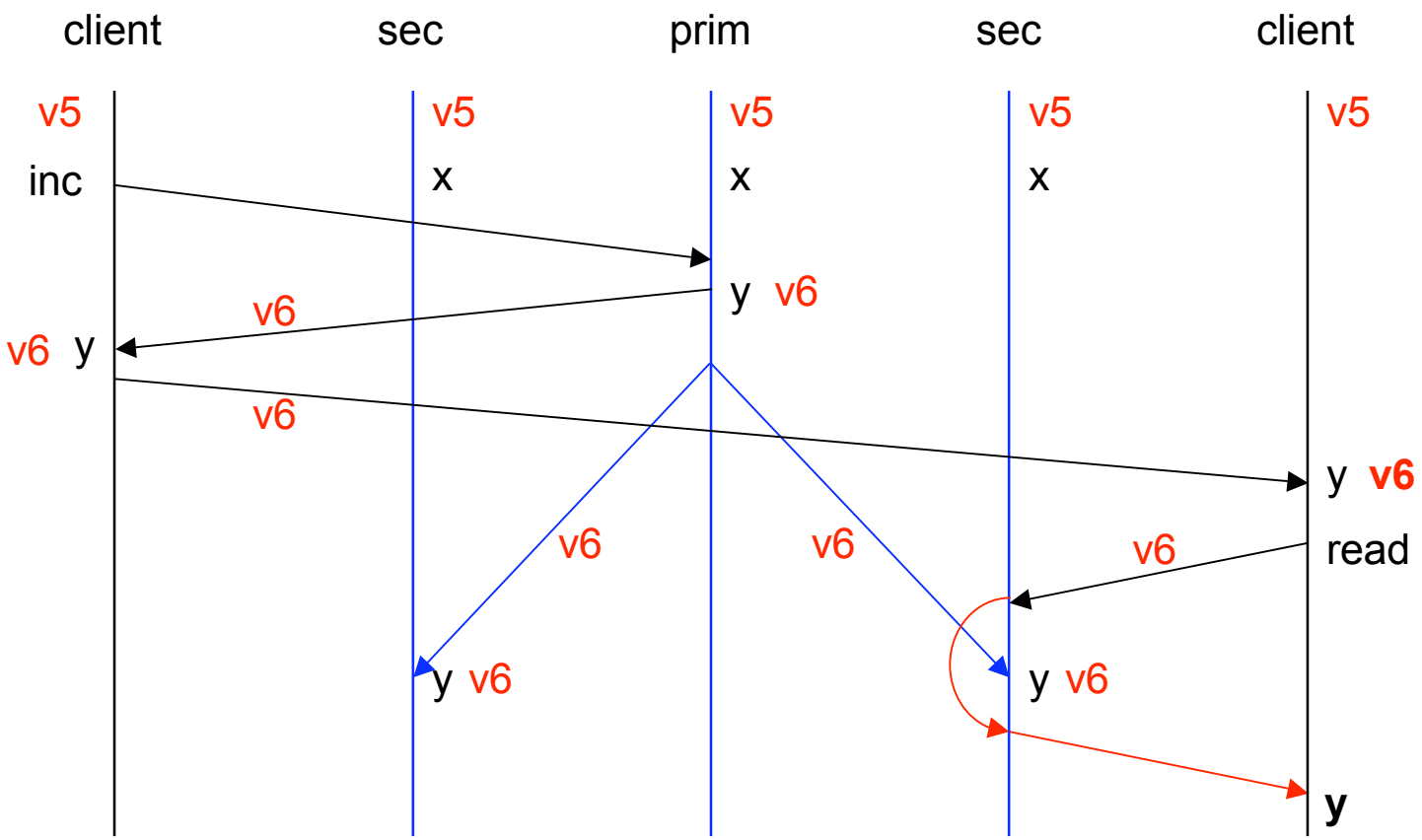
5. Aufgabenblatt

Konsistenz replizierter Objekte

Konsistenz replizierter Objekte

- sequentielle Konsistenz
  - Äquivalenz zu sequentieller Ausführung
  - Implementierung: total geordnete Rundrufe über Koordinator (Primärkopie)
- kausale Konsistenz
  - Implementierung: kausal geordnete Rundrufe
- PRAM-Konsistenz
  - Implementierung: oft schon gegeben (FIFO)
- schwache Konsistenz
  - bei Zugriff auf Synchronisationsobjekte
  - eager release consistency:
    request-/release-Operationen, Abgleich bei release
  - lazy release consistency: Abgleich bei request

Realisierung sequentieller Konsistenz



- Eintrittskonsistenz
  - request/release
  - freigranulare Sperrung, bis zu einem Synchronisationsobjekt pro Datenelement (Freigabekonsistenz legt bei request fest, welche Variablen benötigt werden)
  - höhere Nebenläufigkeit, mehr Verwaltungsaufwand
  - -> realisiert in Orca

Klienten-bezogene Konsistenzmodelle

- eventuelle Konsistenz
  - bei einigen Anwendungen kann man mit zeitweiser Inkonsistenz leben bzw. nimmt sie in Kauf, etwa DNS, Proxy
  - insbesondere, wenn Klient immer auf das gleiche Replikat zugreift
- monotone Lese-Konsistenz
  - Klient liest das gleiche Datenelement immer in gleicher oder zunehmend aktueller Version
- monotone Schreib-Konsistenz
  - Klient beobachtet Schreiboperationen eines Prozesses in der korrekten Reihenfolge
  - Unterschied zu FIFO/PRAM:
    Forderung für den Klienten (Leser) nicht für das Replikat/verteilte Objekt

- read-your-writes-Konsistenz
  - Klient liest immer den von ihm zuletzt geschriebenen Wert (sofern keine anderen späteren Schreiboperationen)
- writes-follow-read-Konsistenz
  - Schreiboperation findet auf einer Kopie statt, die mindestens so aktuell ist, wie die Quelle der letzten Leseoperation
  - z.B. Antwort auf Newsgroup-Artikel (dann keine weitere Forderung für Leser)