Strong Invariants for Weak Consistency

Gustavo Petri
Marc Shapiro
Masoud Saeida-Ardekani

Consistency & Invariants

- Consistency in 3D
  - Characterization of consistency models according to the guarantees they provide
- Dimensions of Guarantees
  - Single object
  - Propagation of effects on different objects
  - Composition of objects

How much can I get for free?

Which invariants are guaranteed by the consistency model **without** additional instrumentation?

Three classes…

	…of invariant	… of protocol
Gen1	Constrain value of an object	Total order of operations
PO	Ordering between operations	Visibility
EQ	State equivalence between objects	Composition

Consistency in 3D

Total Order Axis (Gen1)
How Operations on Individual Objects are Updated/Observed

$\{\ 0 \leq balance \leq \texttt{MAX\_INT}\ \}$

Partial Order Axis (PO)
How Operations on Different Objects are Updated/Observed

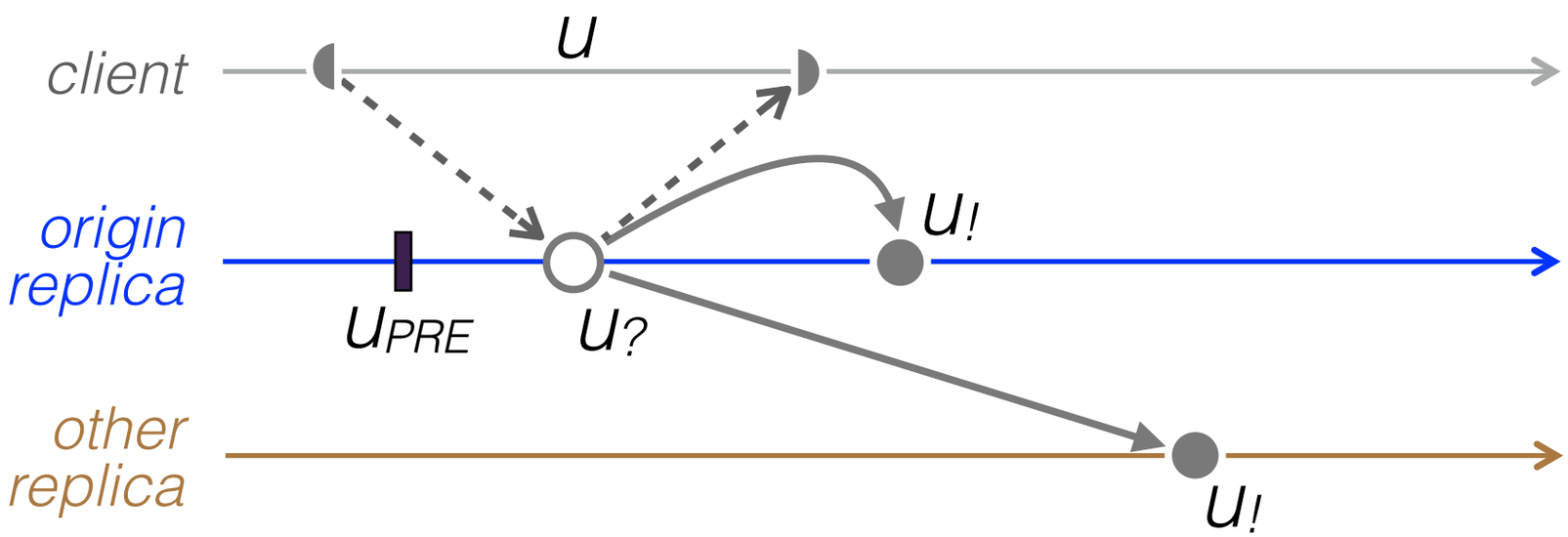
$\{\ x \leq y\ \}$

Equality Axis (EQ)
How Composed Operations on Different Objects are Updated/Observed

$\{\ x \in \mathsf{friendsOf(y)} \iff y \in \mathsf{friendsOf}(x)\ \}$

Program Model: Operationally

Operation



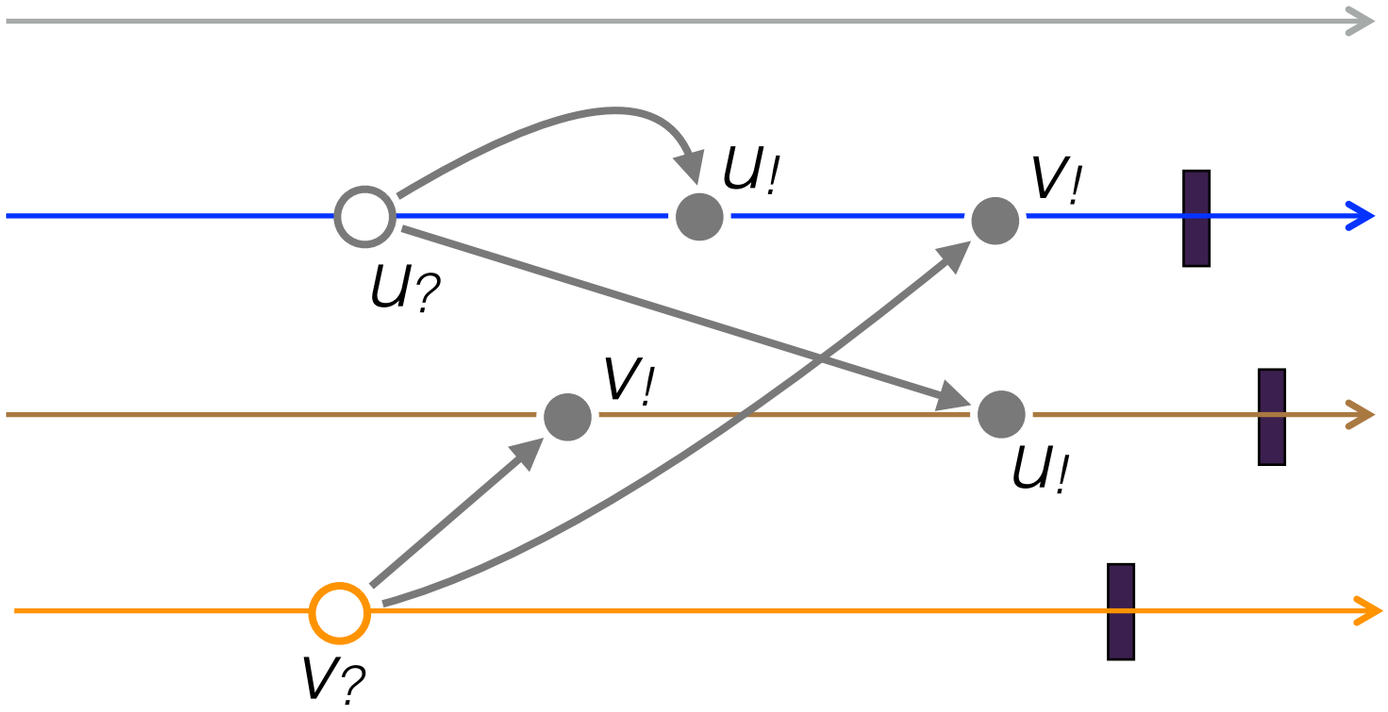
u: state ⇝ (retval, (state ⇝ state))

Prepare (@origin) *u?*; deliver *u!*

Read one, write all (ROWA)

Deferred-update replication (DUR)

Concurrent



Concurrent, Multi-master

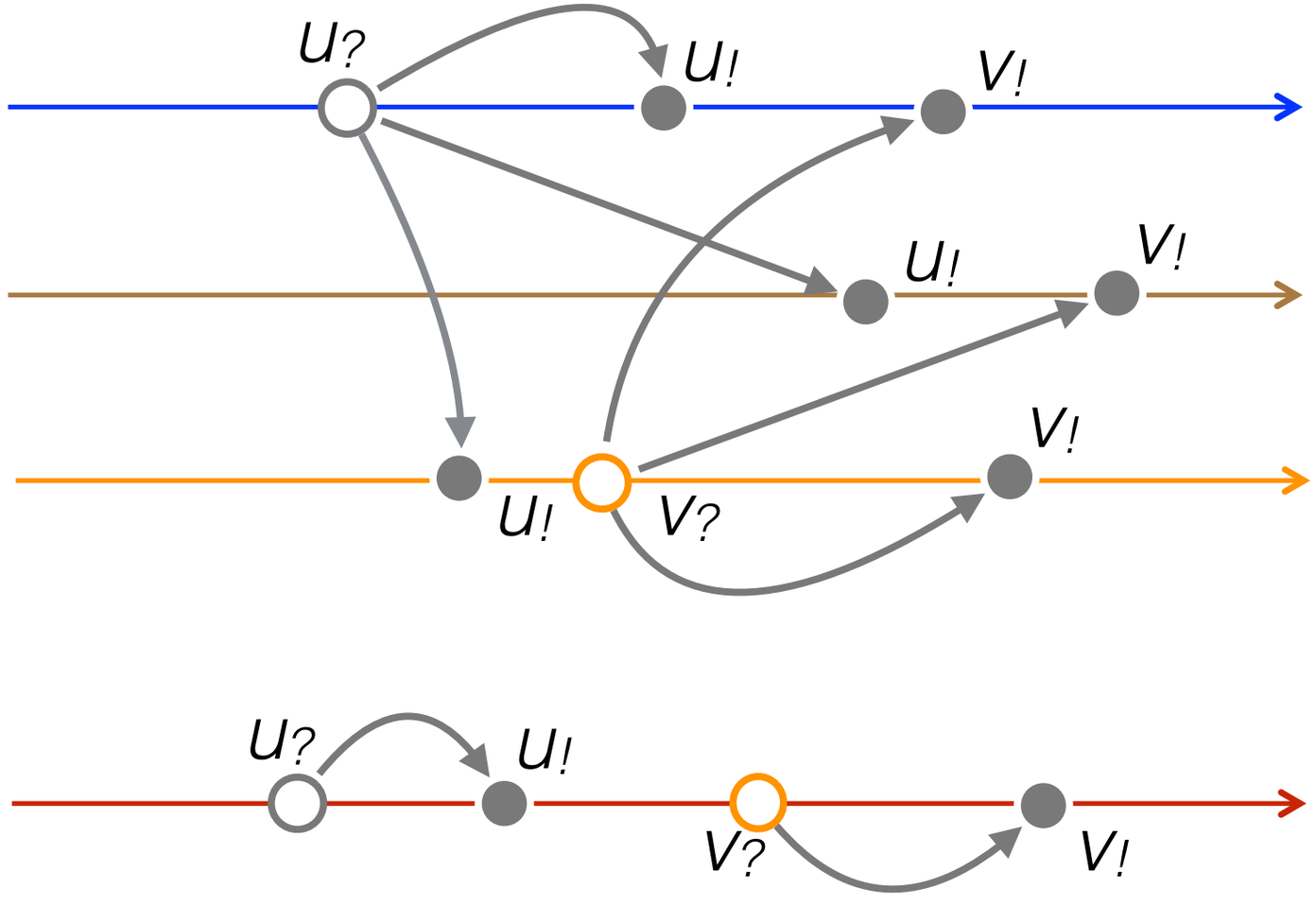
Strong: total order, identical state

Weak: concurrent, interleaving, no global state

Axiomatic definitions can be derived from the operational ones

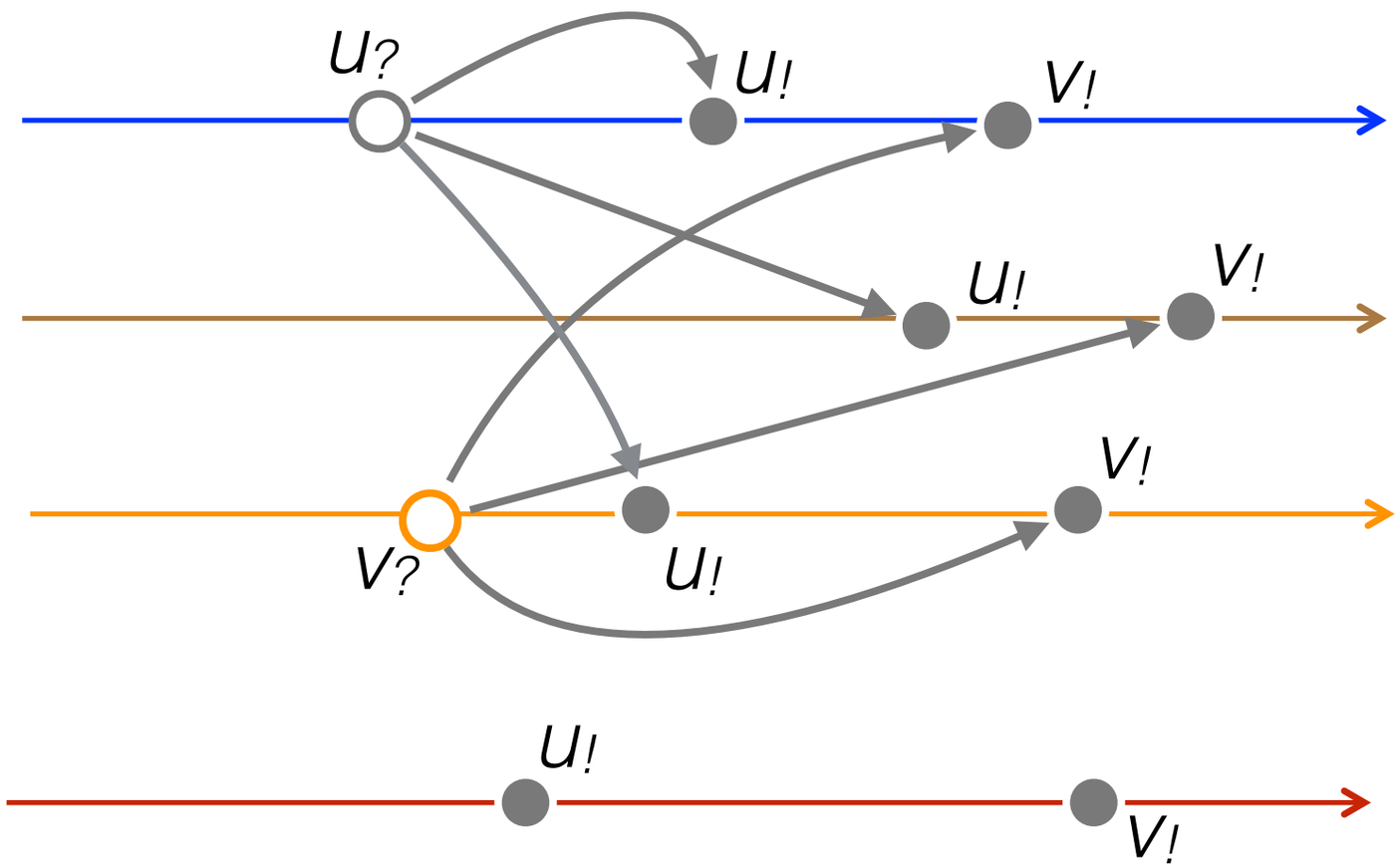
Total Order Axis

- Assumption: Single Object
- Total Order of Effectors and Generators (TOE=TOG)



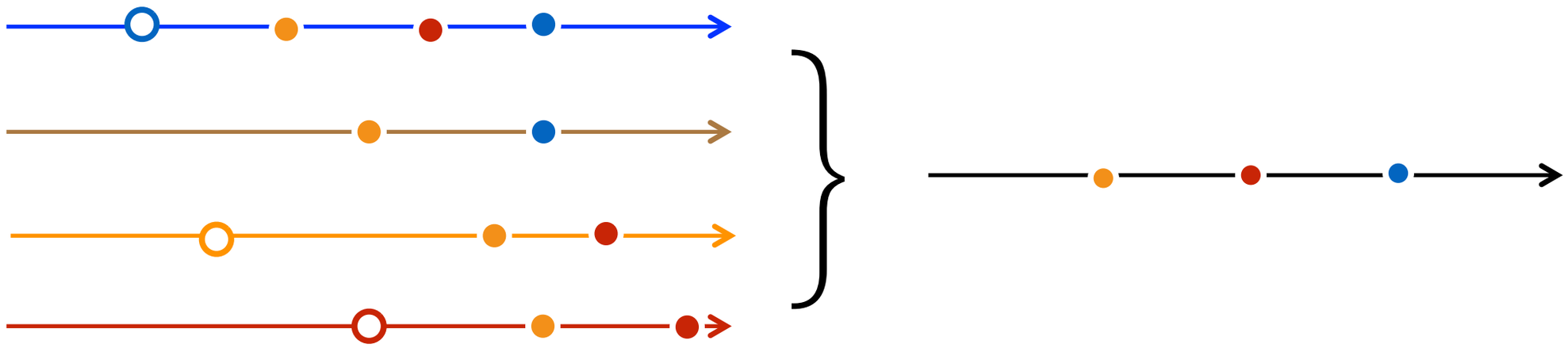
Total Order Axis

- Assumption: Single Object
- Total Order of Effectors ~~and Generators~~ (TOE_1)



Total Order Axis

- Assumption: Single Object
- Total Order of Effectors ~~and Generators~~ (TOE_1)
  - Gapless TOE_1: all replicas apply all effectors in the same order
  - Capricious TOE_1: replicas apply a subset of the effectors in an order consistent with a global total order



- Concurrent Updates (No Global Ordering)

(TO)

Total order updates + queries

Negotiated total order updates

Total order, capricious

Concurrent

Total Order Axis (Gen1)

- Assumptions:
  - (i) Single Object,
  - (ii) State Based,
  - (iii) O is a valid object for *I* [eg. Owicki/Gries proof]

Lemma (TOE=TOG ⇒ Same State)**.** *Any execution on a single object satisfying the TOE=TOG axioms is equivalent to a Sequential execution of the operations.*

Lemma (TOE=TOG ⇒ Linearizable)**.** *Any execution on a single object satisfying the TOE=TOG axioms is equivalent to a Sequential execution of the operations.*

Corollary (TOE=TOG ⇒ Gen1)**.** *Let O be a valid object w.r.t. the invariant I. Any execution of the Most General Client of O under a model satisfying the TOE=TOG axioms satisfies I.*

Gapless TOE

- Assumptions:
  - (i) Single Object,
  - (ii) State Based,
  - (iii) O is a valid object for I [eg. Owicki/Gries proof]
- Release Acquire (RA) Memory Model [Lahav&Vafeiadis'15]

Definition 4. A relation R is called a *modification order* for a location $x \in \mathsf{Loc}$ in an execution G if the following hold: (i) R is a total strict order on $\mathtt{W}_x \cup \mathtt{U}_x$; (ii) if $\langle a, b\rangle \in E^*_{all}$ then $\langle b, a\rangle \notin R$; (iii) if $\langle a, b\rangle \in E^+_{all}$ and $\langle c, b\rangle \in E_x$ then $\langle c, a\rangle \notin R$; and (iv) if $\langle a, b\rangle, \langle b, c\rangle \in R$ and $c \in \mathtt{U}$ then $\langle a, c\rangle \notin E_x$.

Lemma (Gapless TOE $\sim$ RA)**.** *Any execution on a single object satisfying the TOE axioms is equivalent to an RA execution of the operations.*

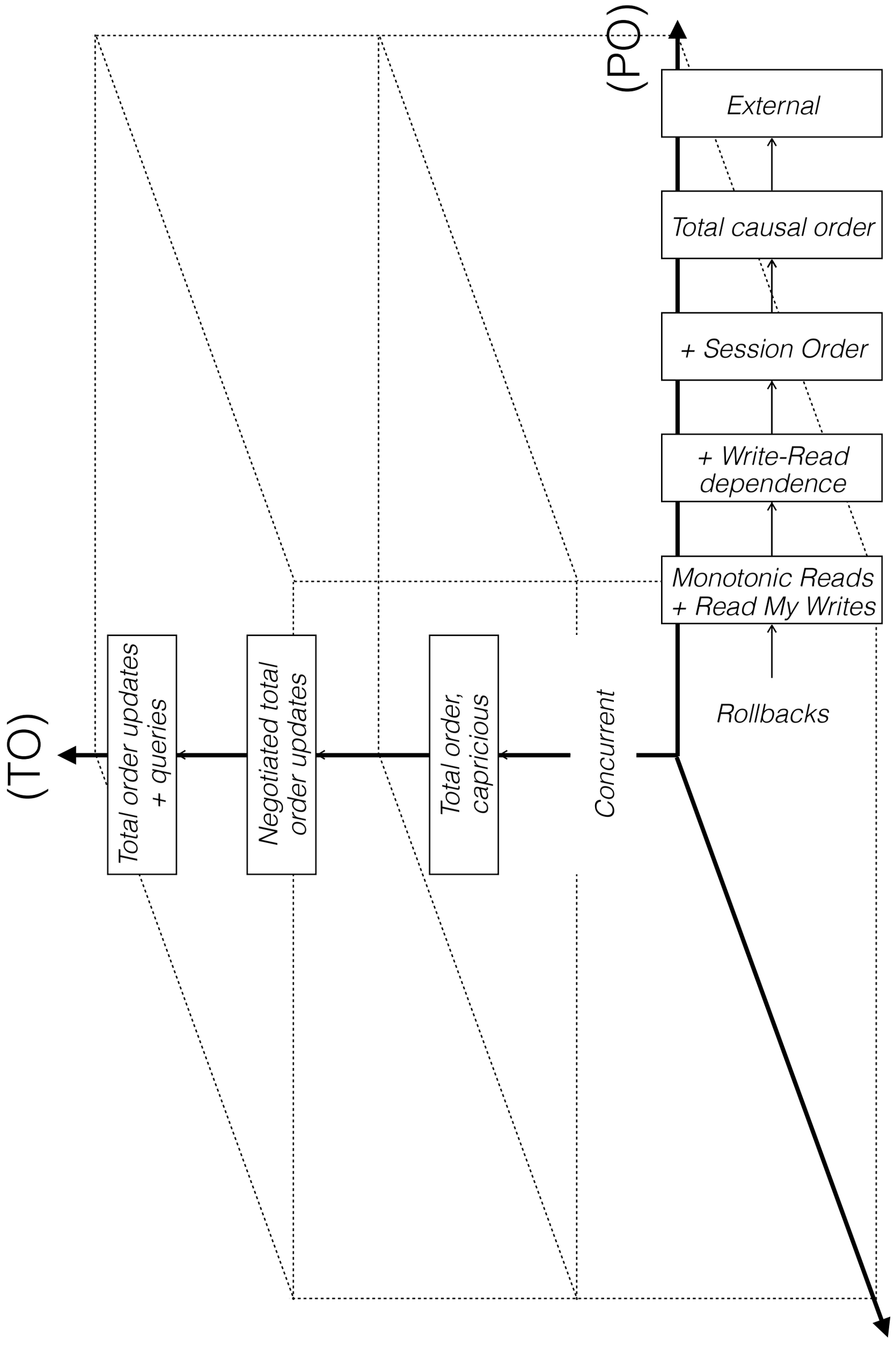
Corollary (OGRA for Gapless TOE)**.** *The OGRA logic of [Lahav& Vafeiadis'15] is sound for models satisfying the Gapless TOE axioms.*

Partial Order Axis

- Assumption: Multiple (2) Objects

- Client Guarantees:
  - Read Own Writes
  - Monotonicity (Reads/Writes)
  - Preservation of (anti)Dependencies

- Visibility Properties:
  - Transitive Visibility
  - Causal Visibility


(PO)
External
Total causal order
+ Session Order
+ Write-Read dependence
Monotonic Reads + Read My Writes
Rollbacks
Concurrent
(TO)
Total order updates + queries
Negotiated total order updates
Total order, capricious

Partial Order Axis (Invariants)

- Assumptions:
  - (i) Multiple Object,
  - (ii) State Based,
  - (iii) O is a valid object for I

- Invariants Relating Objects
  - $x \leq y$
  - $P(x) \Longrightarrow Q(y)$

- Programming:
  - Demarcation Protocol
  - Escrow

Demarcation Protocol

VLDB Journal, 3, 325-353 (1994), Peter Scheurermann, Editor

The Demarcation Protocol: A Technique for Maintaining Constraints in Distributed Database Systems

Daniel Barbará-Millá and Hector Garcia-Molina

Received August, 1992; revised version received June, 1993; accepted June, 1993.

Abstract. Traditional protocols for distributed database management have a high message overhead; restrain or lock access to resources during protocol execution; and may become impractical for some scenarios like real-time systems and very large distributed databases. In this article, we present the demarcation protocol; it overcomes these problems by using explicit consistency constraints as the correctness criteria. The method establishes safe limits as "lines drawn in the sand" for updates, and makes it possible to change these limits dynamically, enforcing the constraints at all times. We show how this technique can be applied to linear arithmetic, existential, key, and approximate copy constraints.

Demarcation Protocol*

$$I = \{\ x \geq y\ \wedge\ (\forall i.\ A_i \geq B_i)\ \}$$

$$\begin{array}{l} x = x + A_1; \\ y = y + B_1; \end{array} \quad \| \quad \begin{array}{l} x = x + A_2; \\ y = y + B_2; \end{array} \quad \| \quad \begin{array}{l} x = x + A_3; \\ y = y + B_3; \end{array}$$

Usual approach: ghost variables

$$I = \{\ x - (\sum \mathsf{ite}(a_i, A_i, 0)) \geq y - (\sum \mathsf{ite}(s_i, B_i, 0))\ \wedge (\forall i.\ A_i \geq B_i)\}$$

$$\left\langle \begin{array}{l} x = x + A_1; \\ a_1 = true; \\ y = x + B_1; \\ s_1 = true; \end{array} \right\rangle \quad \| \quad \left\langle \begin{array}{l} x = x + A_2; \\ a_2 = true; \\ y = x + B_2; \\ s_2 = true; \end{array} \right\rangle \quad \| \quad \left\langle \begin{array}{l} x = x + A_3; \\ a_3 = true; \\ y = x + B_3; \\ s_3 = true; \end{array} \right\rangle$$

* Program Order as communication

Program Order Axis

- Assumptions:
  - (i) Multiple Object,
  - (ii) State Based,
  - (iii) O is a valid object for I

Lemma (CC + $\text{TOE}_1 \sim$ RA)**.** *For any execution satisfying the axioms of* Causal Consistency*, and when projected over a single object satisfies the* TOE axioms *(TOE_1), there exists an equivalent RA execution.*

Corollary (OGRA for CC + TOE_1)**.** *The OGRA logic of [Lahav& Vafeiadis'15] is sound for models satisfying CC and TOE_1.*

Remark (Transitive + $\text{TOE}_1 \not\sim$ RA)**.** *Models satisfying Transitive Visibility instead of Causal visibility are not necessarily equivalent to RA.*

Demarcation Protocol

Template Proof for Demarcation-style Programs**

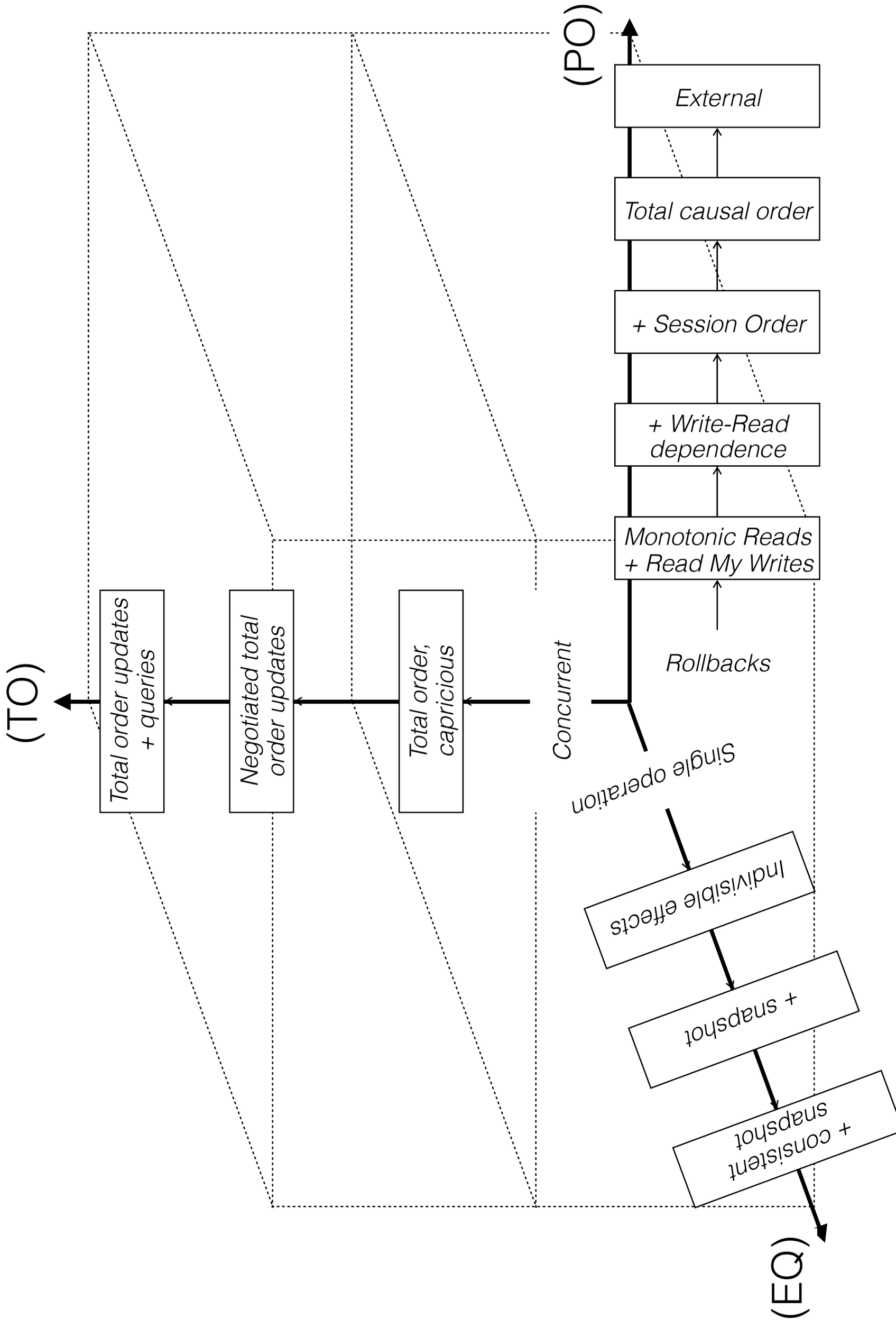
$$I = \{\ x - (\sum \mathsf{ite}(a_i, A_i, 0)) \geq y - (\sum \mathsf{ite}(s_i, B_i, 0))\ \ \wedge (\forall i.\ A_i \geq B_i)\}$$

$$\left\langle \begin{array}{l} x = x + A_1; \\ a_1 = true; \\ y = x + B_1; \\ s_1 = true; \end{array} \right\rangle \quad \| \quad \left\langle \begin{array}{l} x = x + A_2; \\ a_2 = true; \\ y = x + B_2; \\ s_2 = true; \end{array} \right\rangle \quad \| \quad \left\langle \begin{array}{l} x = x + A_3; \\ a_3 = true; \\ y = x + B_3; \\ s_3 = true; \end{array} \right\rangle$$

**[Lahav&Vafeiadis ghosts are compatible but slightly different]

Equality Order Axis

- Assumption: Multiple (n) Objects
- Transactions
  - Write-atomicity: All-or-nothing
  - Read-atomicity: Snapshot
  - Consistent Snapshot


(PO)
External
Total causal order
+ Session Order
+ Write-Read dependence
Monotonic Reads + Read My Writes
Rollbacks
Concurrent
Single operation
Indivisible effects
+ snapshot
+ consistent snapshot
(EQ)
(TO)
Total order updates + queries
Negotiated total order updates
Total order, capricious

Equality Order Axis

- Assumptions:
  - (i) Multiple Object,
  - (ii) State Based,
  - (iii) O is a valid object for I

Lemma (Strict Serializability $\Rightarrow$ OG)**.** *Any model satisfying (i) consistent snapshots, (ii) "external consistency", and (iii) TOE=TOG can be verified using a coarse-grained Owicki/Gries logic, where non-interference is checked at transaction boundaries.c*

Robustness criteria? [Bernardi,Cerone,Gotsman]

Equality Axis

- Rely Guarantee approach
  - Every Generator/Effector preserves preconditions and the invariant
  - CISE tool [Gotsman et al.'16]

Open Problems & Future Work

- What about operation-based implementations? CRDTs?
- Our characterization of invariants is incomplete