University of Pittsburgh
Computer Science Dept.

Enforcing Policy and Data Consistency of Cloud Transactions

Marian K. Iskander Dave W. Wilkinson
Adam J. Lee Panos K. Chrysanthis

SPCC- ICDCS 2011
The 31st Int'l Conference on Distributed Computing Systems
June 20-24, 2011 Minneapolis, Minnesota, USA

Story of Clouds !

Data replication

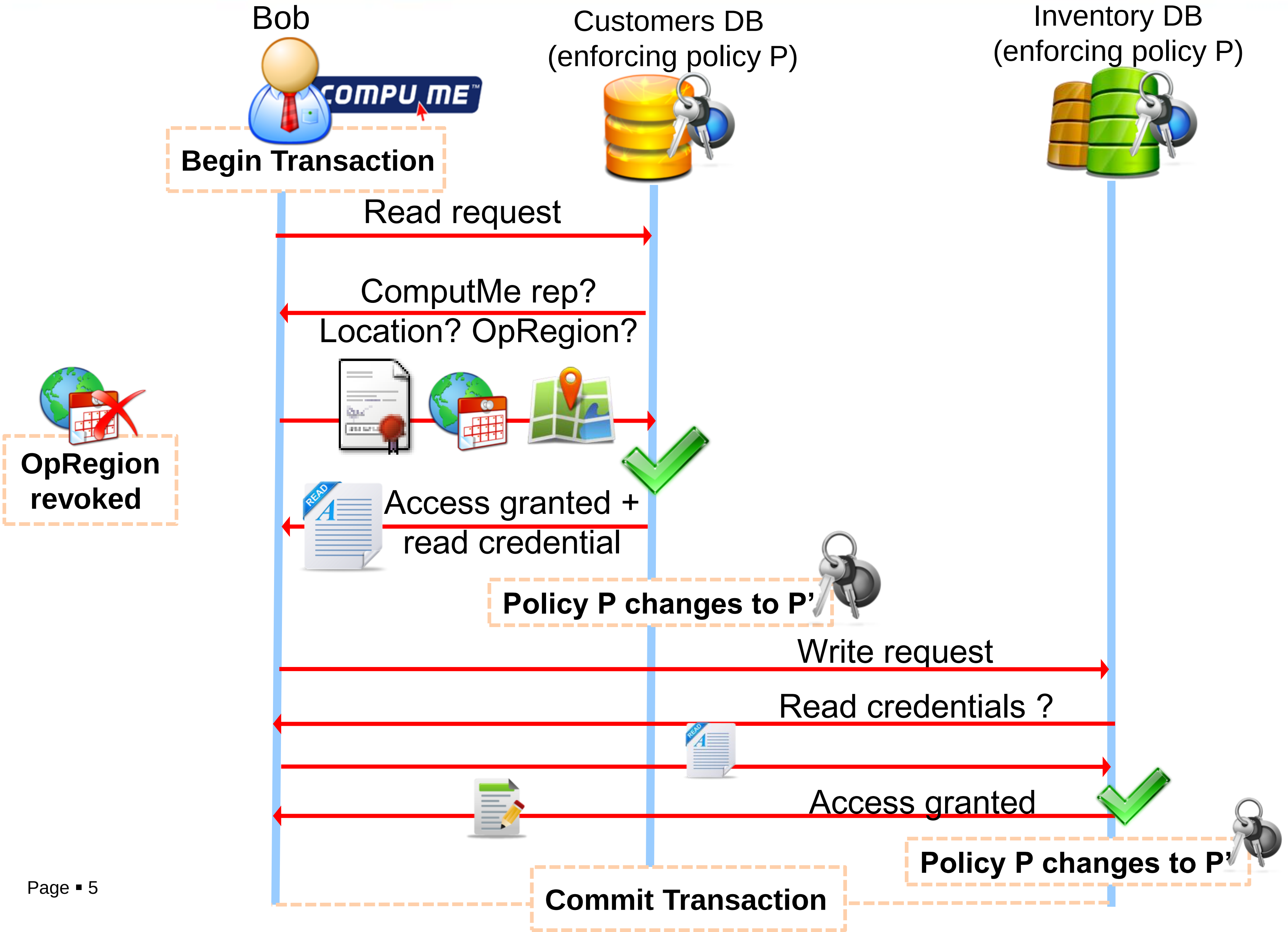
Access control policies replication

Consistency problems

- Data Inconsistency
- Access control policies inconsistency
- User credentials inconsistency [external factors]

Agenda

- Motivating Example
- System Model and Assumptions
- Proofs of authorizations
- Our contributions
  - Consistency Levels
  - Enforcing Trusted Transactions
  - 2PV and 2PVC
- Evaluations
- Conclusions



Bob

Customers DB
(enforcing policy P)

Inventory DB
(enforcing policy P)

Begin Transaction

Read request

ComputMe rep?
Location? OpRegion?

OpRegion revoked

Access granted +
read credential

Policy P changes to P'

Write request

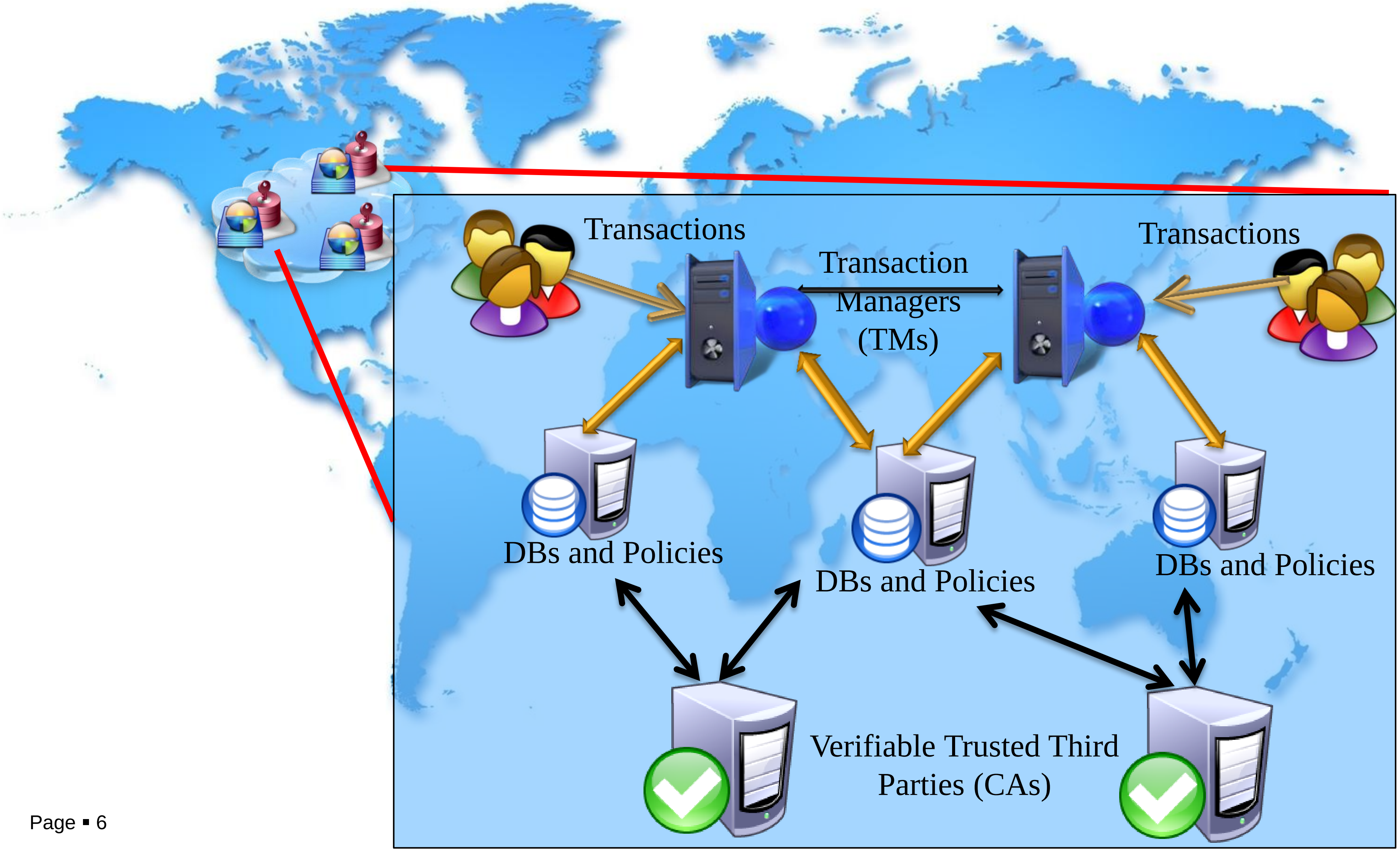
Read credentials ?

Access granted

Policy P changes to P'

Commit Transaction

System model and assumptions



System model and assumptions -- cont

- Credentials:
  - Issued by CAs or by servers (capabilities).
  - Each credential has issuance time and expiration time.
  - Credentials can be prematurely revoked.

- Transactions:
  - Transactions do not fork to sub-transactions.
  - Do not externalize any sensitive data to the users until commit time.

Proofs of authorizations

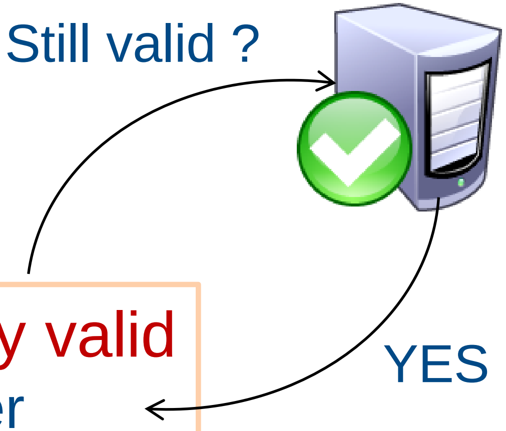
- A proof of authorization is asserted if:

Credential Syntactically valid

Well formed, has valid signature, unexpired

Credential Semantically valid

Unrevoked by issuer



Inference rules are satisfied

Given policy + user credentials

Trusted and Safe Transactions

Trusted Transaction

Satisfies the correctness properties of proofs of authorizations ✓

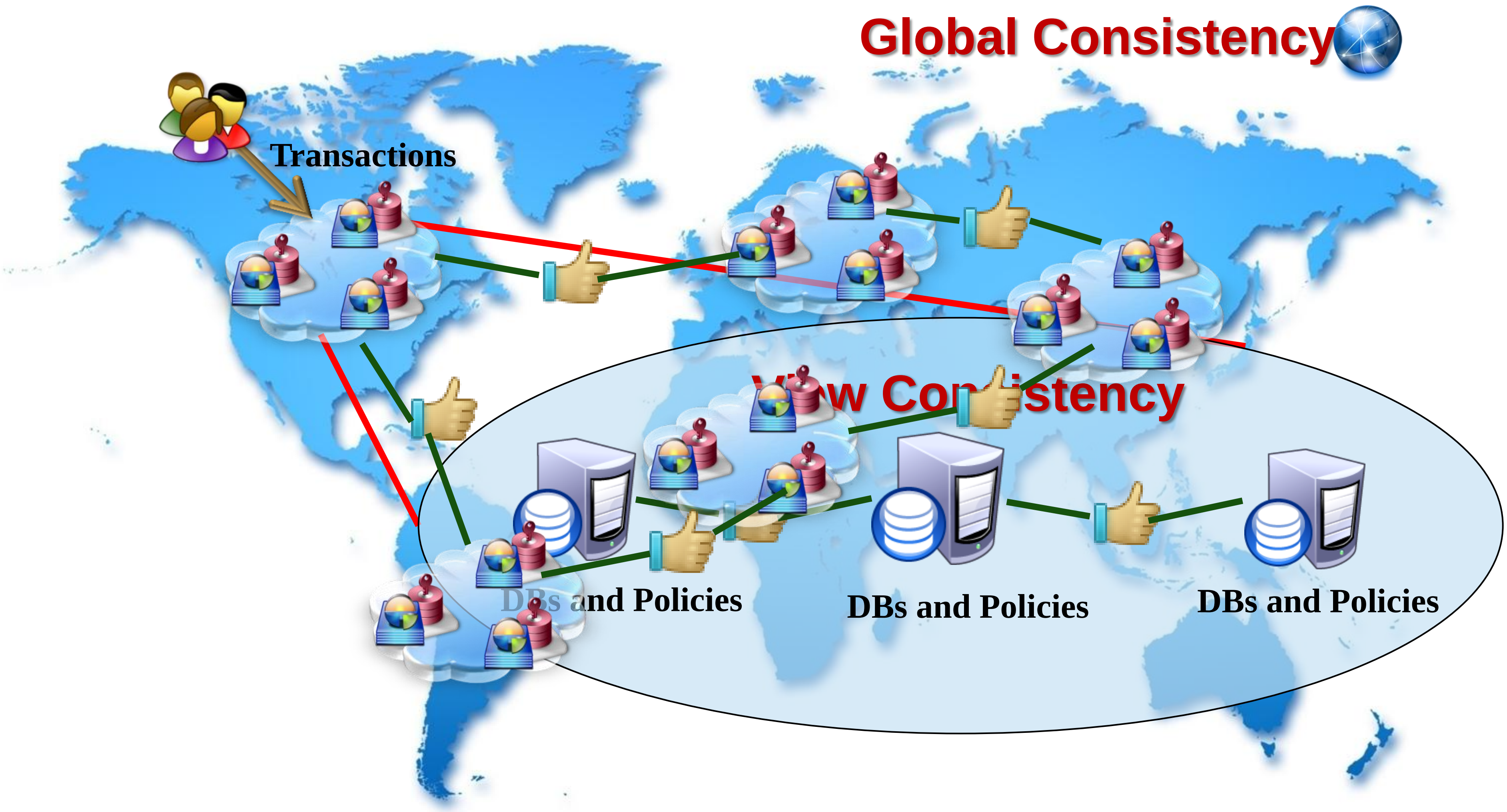
+

Satisfies Data Integrity Constraints

=

Safe Transaction ✓

Consistency Levels



Global Consistency

Transactions

View Consistency

DBs and Policies

DBs and Policies

DBs and Policies

Trusted Transaction

Given a transaction

$T = \{q_1, q_2, . . . , q_n\}$ and its

corresponding view V^T ,

T is trusted iff :

$\forall f_{si} \in V^T : eval(f_{si} , t)$, at

some time instance t :

$\alpha (T) \leq t \leq \omega(T)$

$\wedge (\phi\text{ -consistent}(V^T))$

$\vee (\psi\text{- consistent}(V^T))$

Enforcing Trusted Transactions

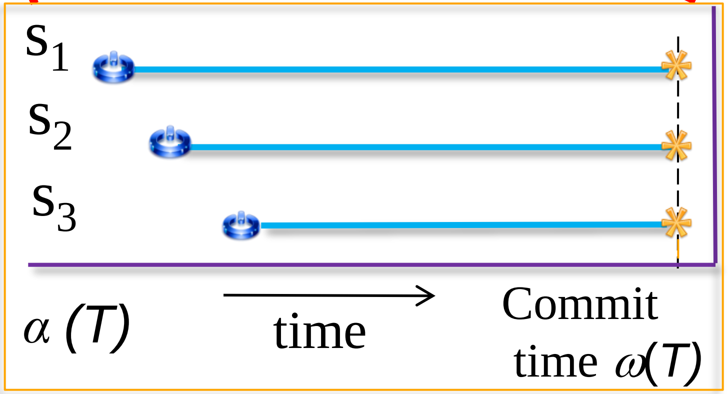
Permissiveness

A. Deferred Proofs of Authorizations

B. Punctual Proofs of Authorizations

C. Incremental Punctual Proofs of Authorizations

D. Continuous Proofs of Authorizations



Properties

- Optimistic
-Most permisive
- Only at commit time when everything is evaluated

Enforcing Trusted Transactions

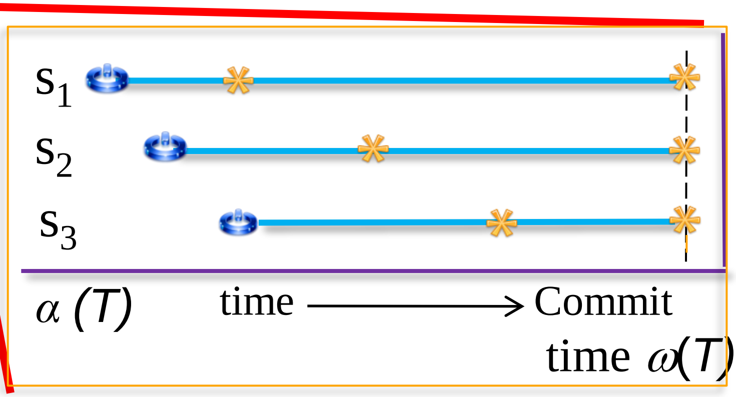
Permissiveness

A. Deferred Proofs of Authorizations

B. Punctual Proofs of Authorizations

C. Incremental Punctual Proofs of Authorizations

D. Continuous Proofs of Authorizations



Properties

- - Proactive
- Possible false positive and fale negative access decisions

Enforcing Trusted Transactions

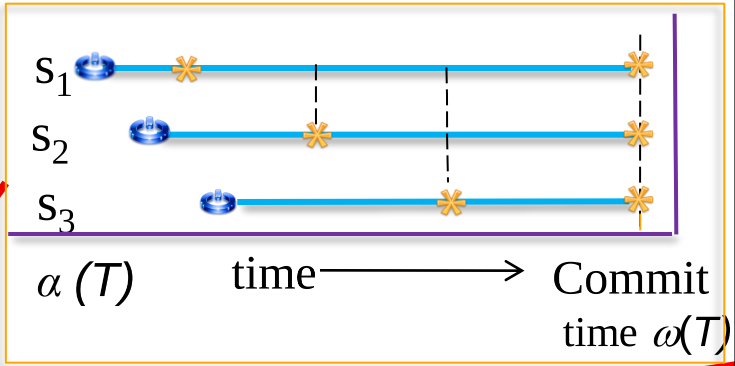
Permissiveness

A. Deferred Proofs of Authorizations

B. Punctual Proofs of Authorizations

C. Incremental Punctual Proofs of Authorizations

D. Continuous Proofs of Authorizations



Properties

- Achieves the desired level of consistency at each server

Enforcing Trusted Transactions

Permissiveness

A. Deferred Proofs of Authorizations

B. Punctual Proofs of Authorizations

C. Incremental Punctual Proofs of Authorizations

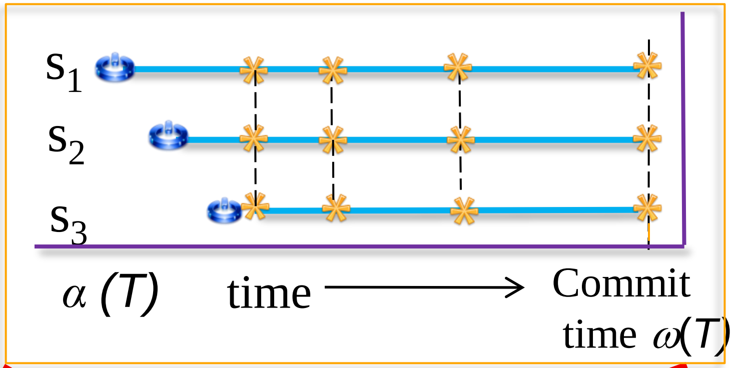
D. Continuous Proofs of Authorizations

Properties

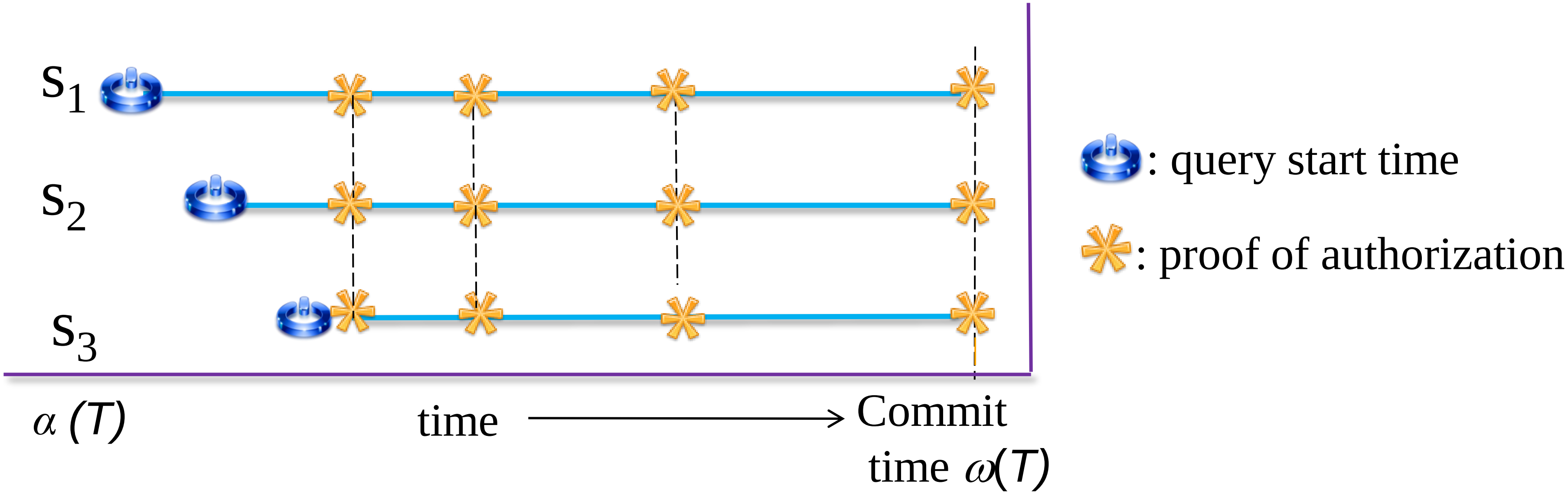
-Least permisive

- Stronger guarantees at any given time

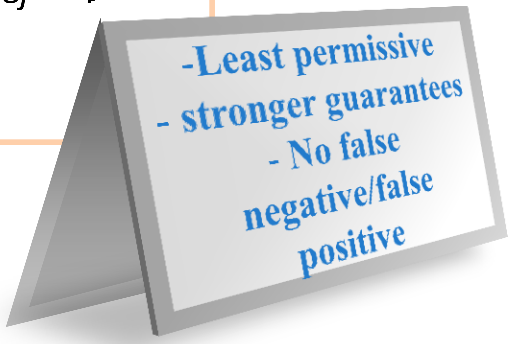
- No false negative/false positive



D- Continuous Proofs of Authorizations



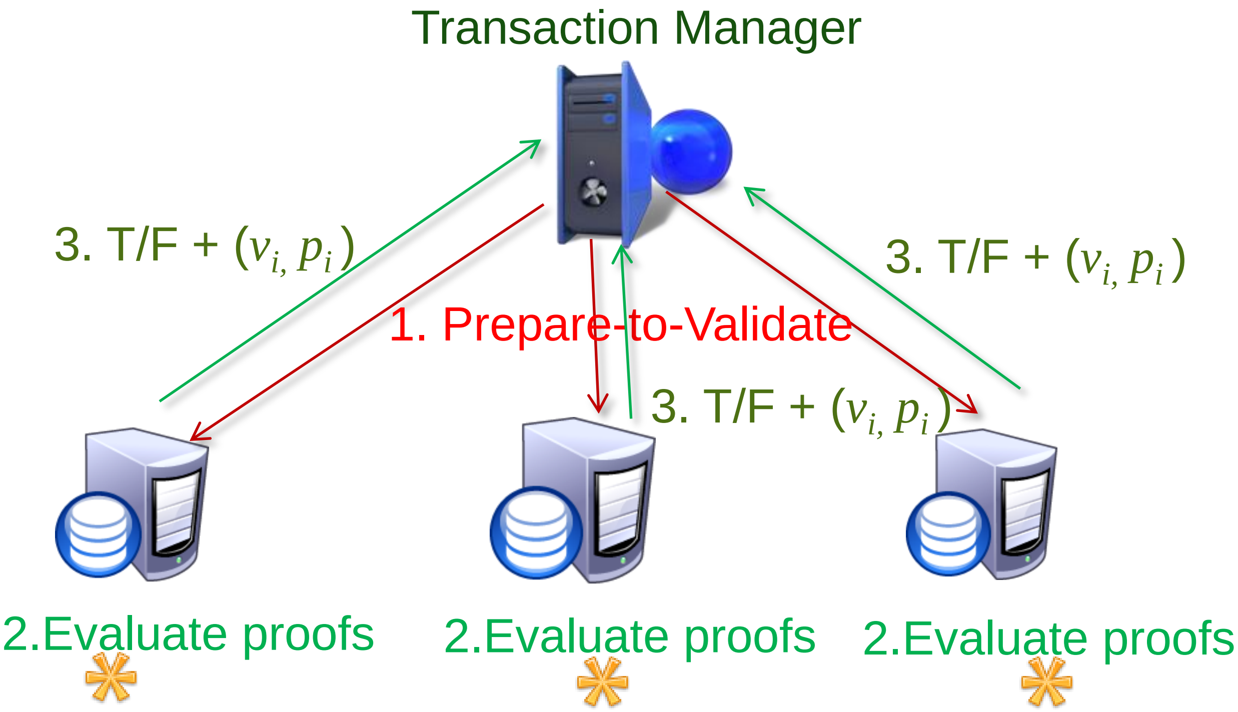
A transaction *T is* declared trusted under the *Continuous approach*, iff ∀ $1 \leq i \leq n$ $\forall 1 \leq j \leq i$: *eval*(f_{si} ,t_i) ∧ *eval*(f_{sj} , t_i) ∧ (ϕ-consistent(V^{T}_{ti}) ∨ ψ-consistent(V^{T}_{ti})) at any time instance t : $\alpha(T) \leq t_i \leq \omega(T)$



-Least permissive
- stronger guarantees
- No false negative/false positive

Two Phase Validation (2PV)

Collection Phase



Validation Phase

- Evaluate responses
- Makes Continue or Abort decision
- In case of Inconsistency → send updates
- Go back to Collection Phase

2PV provides trusted transactions only, what about safe transactions?

Trusted and Safe Transactions

Trusted Transaction ✓ + Data Integrity Constraints = Safe Transaction ✓

2PV protocol + 2PC protocol = 2PVC protocol

Complexity Evaluation

	Deferred	
	View	Global
Messages	$2n + 4n$	$2n + 2nr + r$
Proofs	$2u – 1$	ur

	Punctual	
	View	Global
Messages	$2n + 4n$	$2n + 2nr + r$
Proofs	$u + 2u – 1$	$u+ ur$

	Incremental	
	View	Global
Messages	$4n$	$4n + u$
Proofs	u	u

	Continuous	
	View	Global
Messages	$u (u +1) + 4n$	$u (u +1) + u + 2n + 2nr + r$
Proofs	$u(u+1) / 2$	$u(u+1) / 2 + ur$

Conclusions

- **Identified** prospective consistency problems that can arise as transactional database systems are deployed on cloud servers
- **Defined** the notions of *trusted* and *safe* transactions,
- **Presented** different proofs of authorizations approaches to achieve trusted transactions.
- **Proposed** Two-Phase Validation Commit (2PVC) protocol, an enhanced version of the widely used Two-Phase Commit (2PC) protocol
- **Evaluated** each approach in terms of the performance and applicability.

Thank You
Questions ?

“Consistency is contrary to nature, contrary to life. The only completely consistent people are dead ” **Aldous Huxley**

This work was supported in part by the National Science Foundation under awards CCF–0916015, CNS–0964295, CNS–1017229 and IIS–1050301