Formal Analysis of the CORBA CSIv2 Protocol

Polar Humenn, Susan Older, Shiu-Kai Chin
Systems Assurance Institute
Syracuse University
Syracuse, NY 13244-4100
{polar,sbolder,skchin}@syr.edu

Introduction

- Distributed Security
  - Strong cryptographic algorithms
  - Protocols
  - Trust assumptions
- CORBA
  - Common Secure Interoperability Version 2
- Formal Calculus
  - Papers by Abadi, Lampson, Wobler, Burrows

CORBA

- Proven Middleware for distributed systems
- CSIv2, which is the adopted standard protocol, is a security protocol that includes delegation and authorization information

Agenda

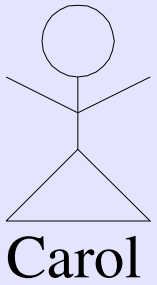
- Explanation of CSIv2
- Calculus for expression complex principals
- Describe a mathematical model of the calculus
- Formal language for expressing logical statements about principals
- Mathematical model for the language
- Example

CSIv2

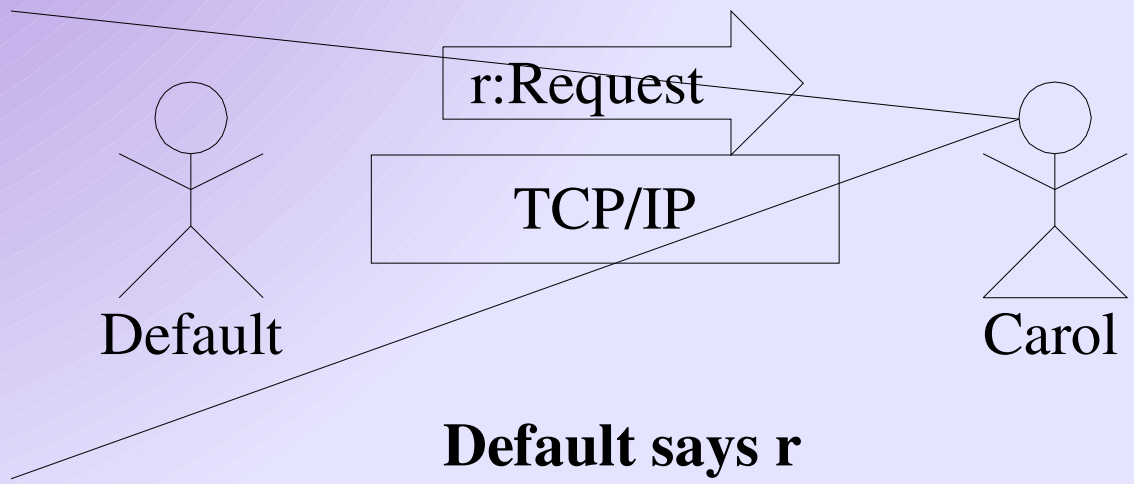
- Passes CORBA Requests with security information.
  - Over Connection-based transport protocol
    - Unauthenticated (e.g. TCP/IP, anonymous SSL)
    - Authenticated (e.g. SSL with certificate, Kerberos).
  - CORBA request payload contains
    - Client Authenticator (AS)
      - Username/Password (GSSUP)
    - Identity Information (SAS)
    - Authorization Information (SAS)
      - Not yet standardized

Principals

- There once was a CORBA server named Carol.
- Carol knows who she is, she is Carol.

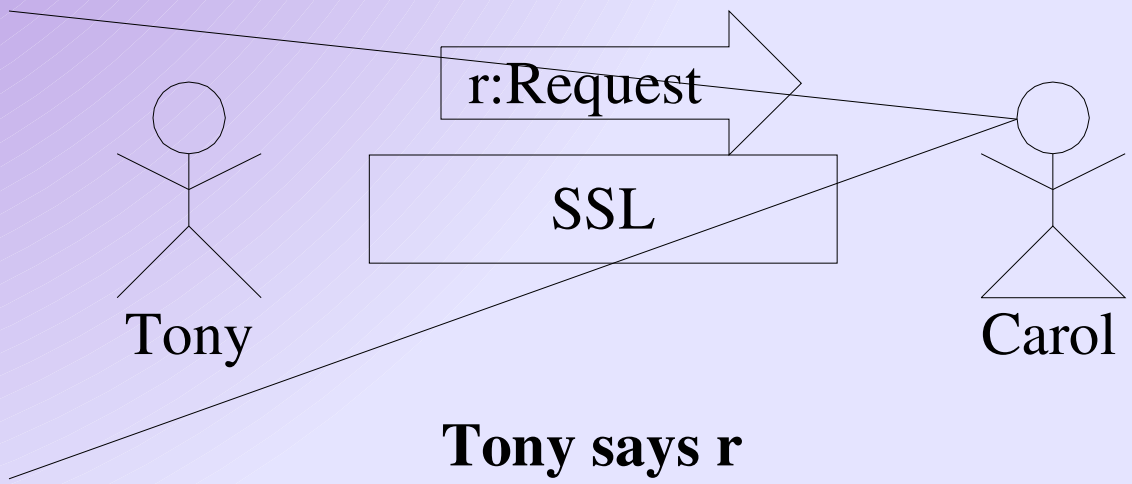


Principals



Default says r

Principals



Tony says r

CSIv2 Principal Structure

- **Transport Layer**
  - Tony or *Default*
- **Request Layer Security Structure <IA,CA,SA>**
  - CA: Client Authentication Layer
    - <Clyde,PW>
  - IA: Identity Token
    - Alex or *Anon*
  - SA: Authorization Information
    - N/A

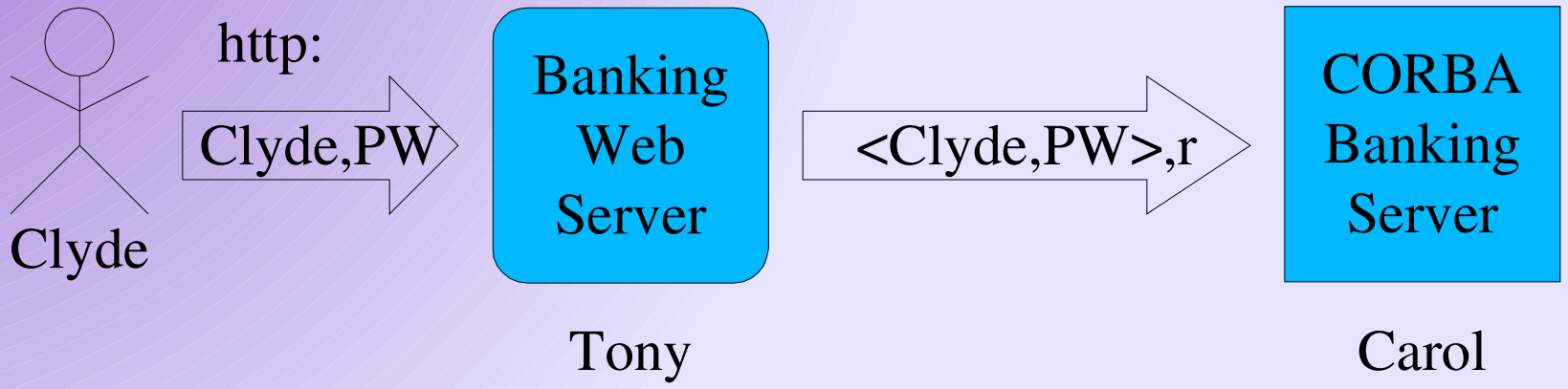
TCP/IP CSIv2 Interpretation

- Default says r
- Default says Anon says r
- Default says Alex says r
- Default says <Clyde,PW> says r
- Default says <Clyde,PW> says Anon says r
- Default says <Clyde,PW> says Alex says r

SSL CSIv2 Interpretation

- Tony says r
- Tony says Anon says r
- Tony says Alex says r
- Tony says <Clyde,PW> says r
- Tony says <Clyde,PW> says Anon says r
- Tony says <Clyde,PW> says Alex says r

Example



- Clyde delegates his authority of access by giving his password to the web server.
- Using CSIv2, the web server, Tony, delivers the authentication information to the CORBA banking server.
- Tony says <Clyde,PW> says r

Formal Reasoning

- Reasoning about the delegation of authority
  - precisely
  - accurately, and
  - consistently

Calculus of Principals

- Assume countable set **Name** ranged over meta-variable *A*.
- Assume a set **PrinExp** ranged over by variables *P* and *Q*, and structured by the following:

$$
\begin{array}{rcl}
P & := & A \\
 & | & P \mid Q \\
 & | & P \wedge Q \\
 & | & P \ \mathbf{for}\ Q
\end{array}
$$

Semantics

- $P := P \mid Q$
  - *P quoting Q*
- $P \wedge Q$
  - *P* and *Q* agree
- *P* **for** *Q*
  - *P* is authorized to speak on *Q*'s behalf

Intuition

- P /\ Q is stronger than P or Q alone because of the consensus.
- P for Q is stronger than P | Q because of the authorization.
- Support this intuition by finding a model of the semantics using a multiplicative semi-lattice semi-group (MSS).

MSS

- A multiplicative semi-lattice semi-group is a mathematical structure, <S,⩏,||>, in which a set, S, is organized with the two binary operators ⩏ and ||. The ⩏ operator must be idempotent, associative, and commutative. The || operator must be associative and must distribute over the ⩏ operator in both arguments. Specifically, for all elements x, y, z ∈ S, the following equations hold:

$$
\begin{aligned}
x \Cup x &= x \\
x \Cup y &= y \Cup x \\
x \Cup (y \Cup z) &= (x \Cup y) \Cup z) \\
x \parallel (y \parallel z) &= (x \parallel y) \parallel z \\
x \parallel (y \Cup z) &= (x \parallel y) \Cup (x \parallel z) \\
(y \Cup z) \parallel x &= (y \parallel z) \Cup (z \parallel x)
\end{aligned}
$$

- Induces partial order, $x \Cup y$ if and only if $x = x \Cup y$.

Mathematical Semantics

Assume a set, W, and we take a subset of WxW, W_R, such that $\langle W_R, \cup, \circ \rangle$ satisfies the MSS axioms, and we have a function J from **Name** to W_R, we extend J to J' in the following manner:

$$J'(A) = J(A)$$

$$J'(A \wedge B) = J'(A) \cup J'(B)$$

$$J'(A \mid B) = J'(B) \circ J'(A)$$

$$\{ (w,w'') \mid \exists\, (w,w') \in J'(B) \;\&\; (w',w'') \in J'(A) \}$$

Then,

$$J'(A) \leq J'(B) \quad \text{iff} \quad J'(A) \supseteq J'(B)$$

Extending $\leq$ to PrinExp, it follows:

$$A \wedge B \leq A$$

A ∧ B is *below* A in the MSS and therefore A ∧ B is *stronger* than A itself.

What about delegation? A **for** B

- Abadi suggests that A **for** B can be coded as
  - $(A \mid B) \wedge (D \mid B)$

    such that D is a fictional principal that vouches for A authorization to act on B's behalf.
- And if you do the math, we can see that our intuition holds:
  - $(A \textbf{ for } B) \leq (A \mid B)$

Modal Logic

- What is a modal logic?
  - A propositional logic prefixed with a modal operator.
  - P is a principal and φ is a statement in propositional logic.
    - P believes φ
    - P says φ

Logic Syntax

We introduce the set **LogExp** such that:

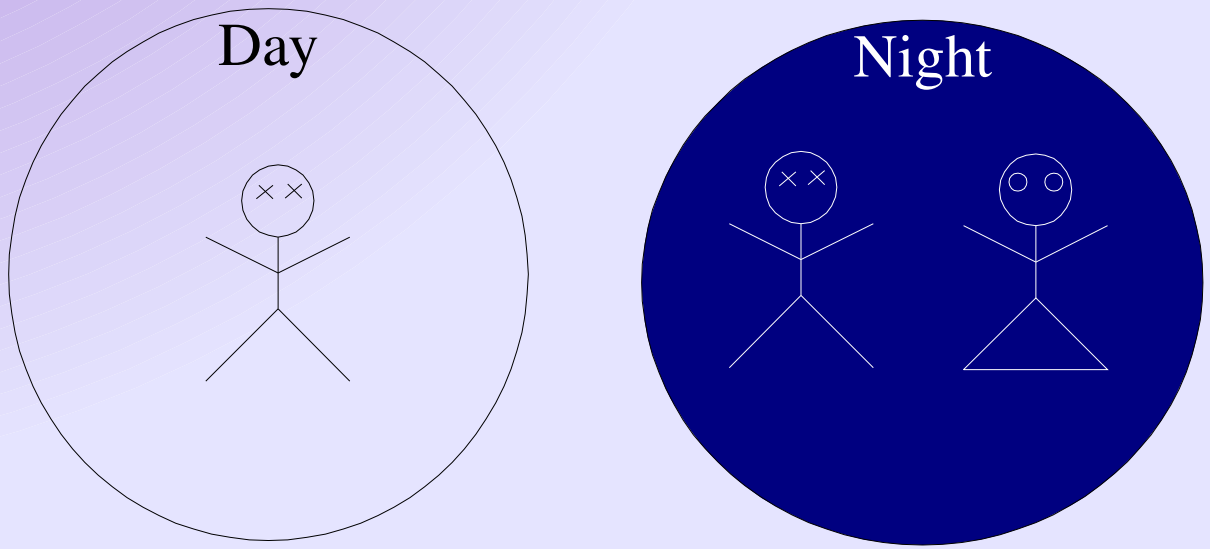
φ :== p
| P **speaksfor** Q
| ~φ
| φ_1 & φ_2
| φ_1 or φ_2
| $\varphi_1 \supset \varphi_2$
| $\varphi_1 \Leftrightarrow \varphi_2$
| P **says** φ

Semantic Axioms

1. $\vdash \varphi$
2. if $\vdash \varphi_1$ and $\vdash \varphi_1 \supset \varphi_2$, then $\vdash \varphi_2$
3. if $\vdash \varphi$, then P **says** φ, for all P
4. $\vdash$ (P **says** ($\varphi_1 \supset \varphi_2$)) $\supset$ ((P **says** φ_1) (P **says** φ_2))
5. $\vdash$ ((P $\wedge$ Q) **says** φ) $\supset$ ((P **says** φ) & (Q **says** φ))
6. $\vdash$ ((P | Q) **says** φ) $\supset$ ((P **says** (Q **says** φ)))
7. $\vdash$ (P **speaksfor** Q) $\supset$ ((P **says** φ) $\supset$ (Q **says** φ)), for all φ

Modal Logic Semantics

- 2 Worlds, Night & Day
- 1 Proposition. It's dark = true.



Semantic Proof

- Kripke structure
  - $M = \langle W, w_0, I, J \rangle$
    - W is a set of worlds
    - w_0 is an initial word in W
    - I is an interpretation of true or false for each propositional variable, p.
    - J is a function that takes a principal and maps it to a set WxW such that $J(P)$ is the set of worlds that P cannot distinguish.
  - ε_M = **LogExp** -> 2^W

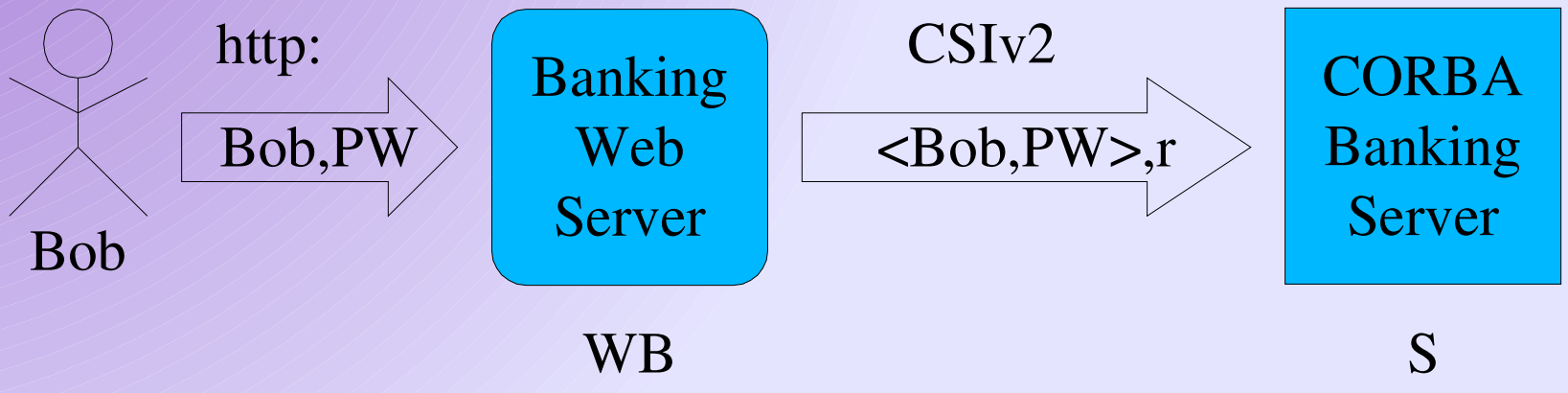
Semantic Conclusion

- From this we write
  - M, w |= φ provided that $w \in \varepsilon_M[\![\varphi]\!]$
  - M |= φ, provided that M, w_0 |= φ
    - M satisfies φ
  - Then, φ is valid if φ is satisfied in all Kripke structures
    - |= φ
  - Finally, |- φ implies that |= φ

Semantic Conclusion

- Our logic is sound with respect to the Kripke semantics, but it is not complete.
  - There are formulas that are valid that are not derivable from the rules we have given.
  - We don't normally consider this too much of a problem.

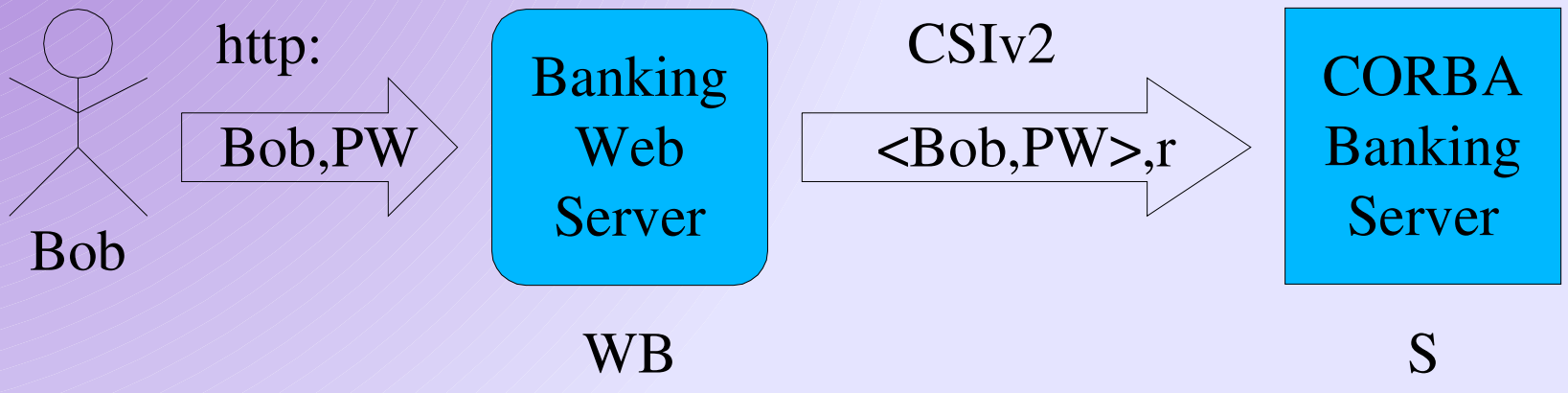
Example



S follows a policy:

WB **for**$_S$ Bob **cando** r

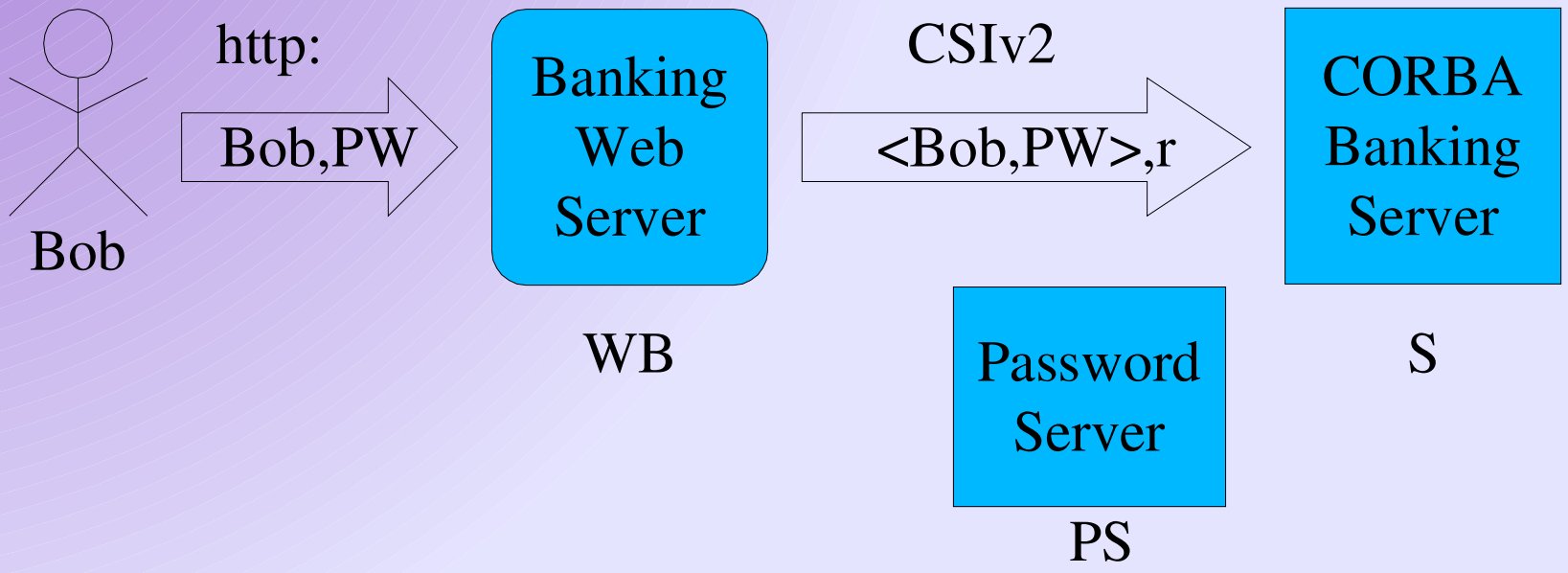
Example



WB using CSIv2 delivers two pieces of information to the banking server, S.

WB **says** Bob **says** r
WB **says** <Bob,PW> **speaksfor** Bob

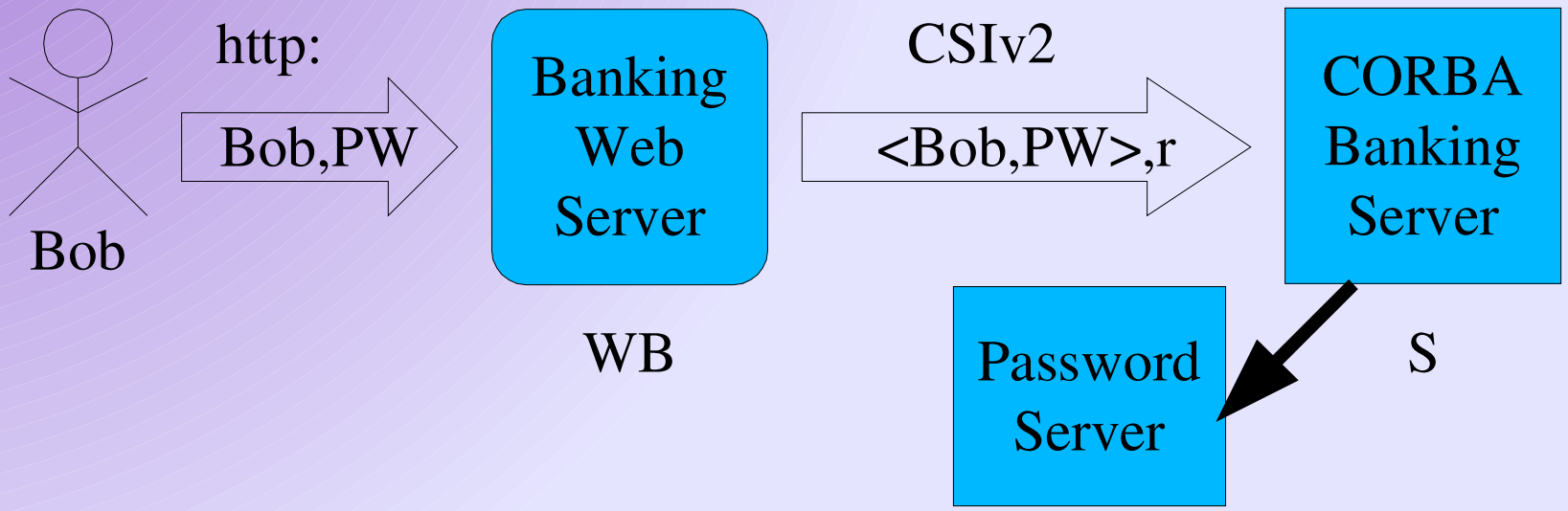
Example



S is governed by the password verification rule:

(WB ∧ PS) says <Bob,PW> speaksfor Bob
⊃ (WB | Bob) speaksfor (S | Bob)

Example

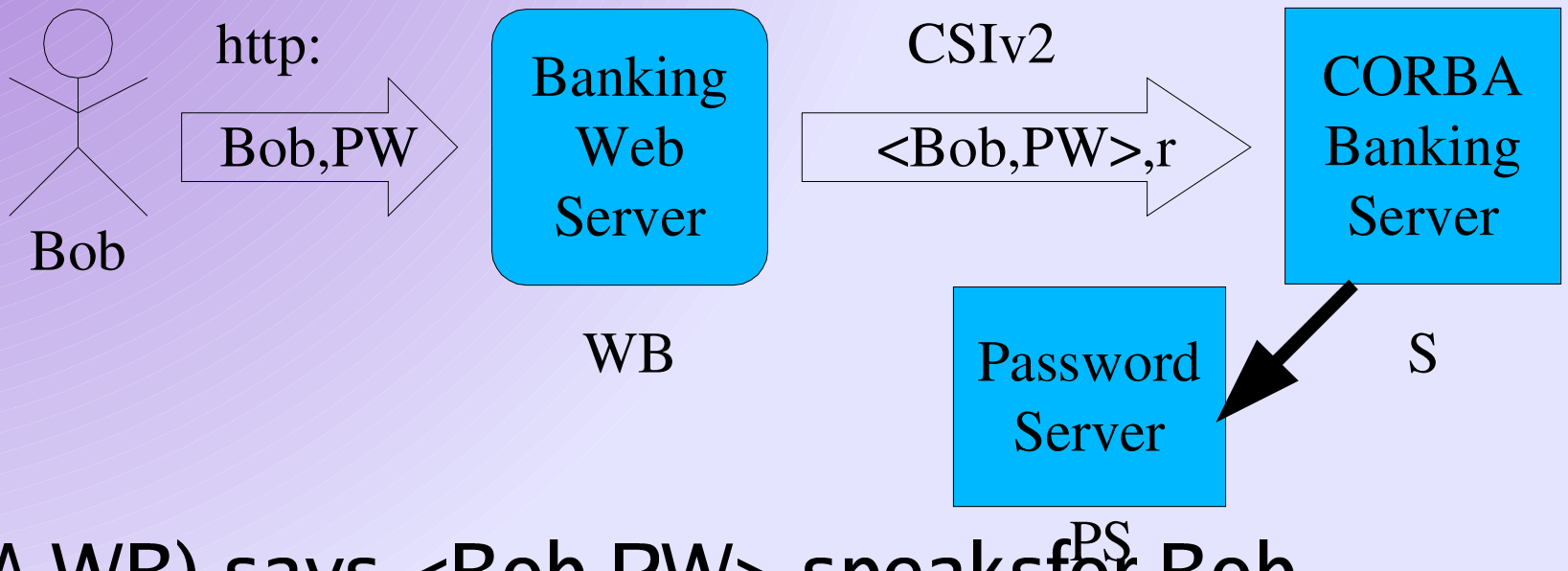


PS says <Bob,PW> speaksfor Bob

WB says <Bob,PW> speaksfor Bob

(PS /\ WB) says <Bob,PW> speaksfor Bob

Example

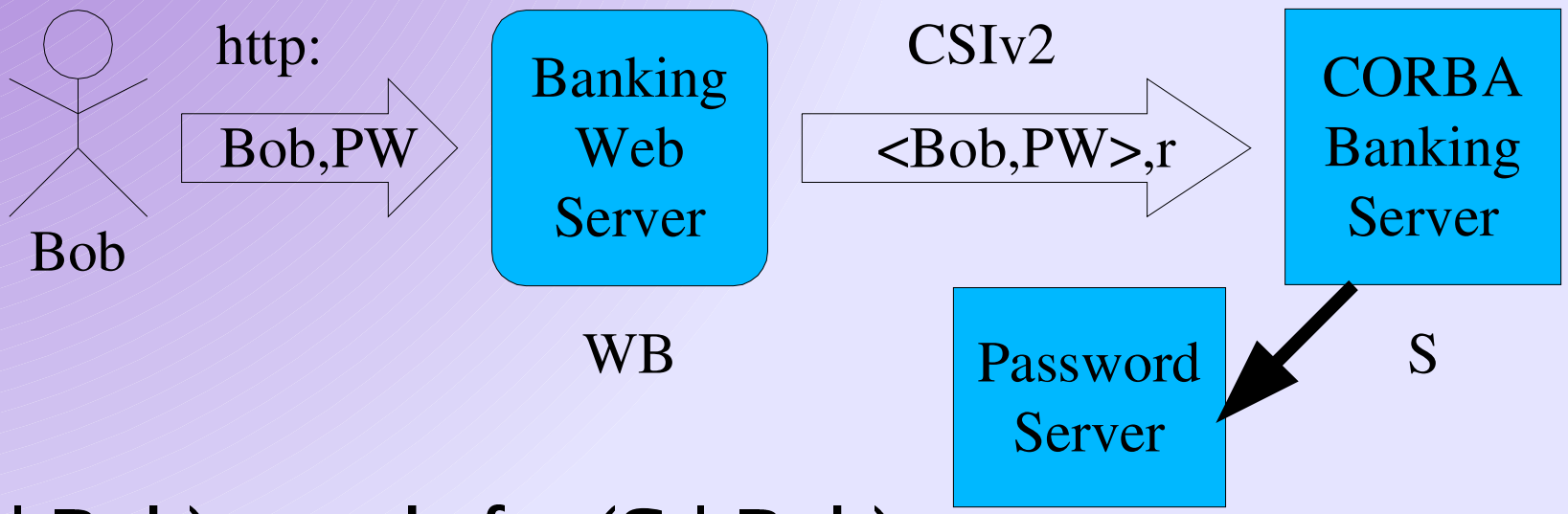


(PS /\ WB) says <Bob,PW> speaksfor Bob

using the password rule

(WB | Bob) speaksfor (S | Bob)

Example

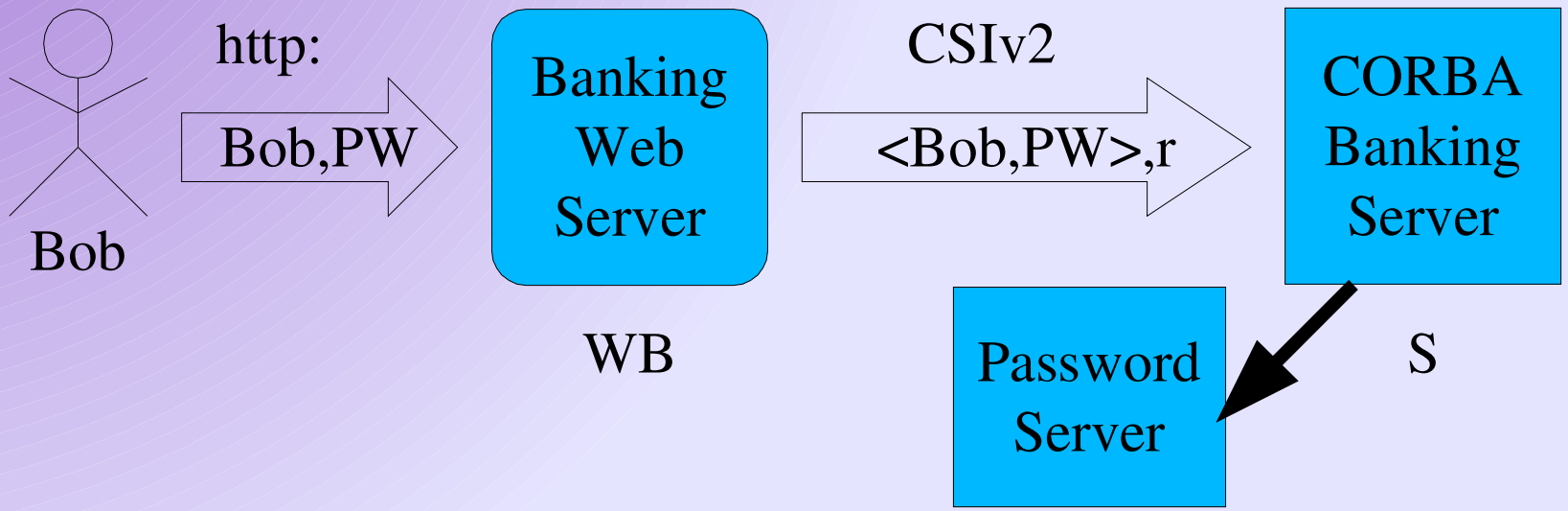


(WB | Bob) speaksfor (S | Bob)

and because speaksfor is a monotonic operator

(WB | Bob) speaksfor (WB | Bob) ∧ (S | Bob)

Example

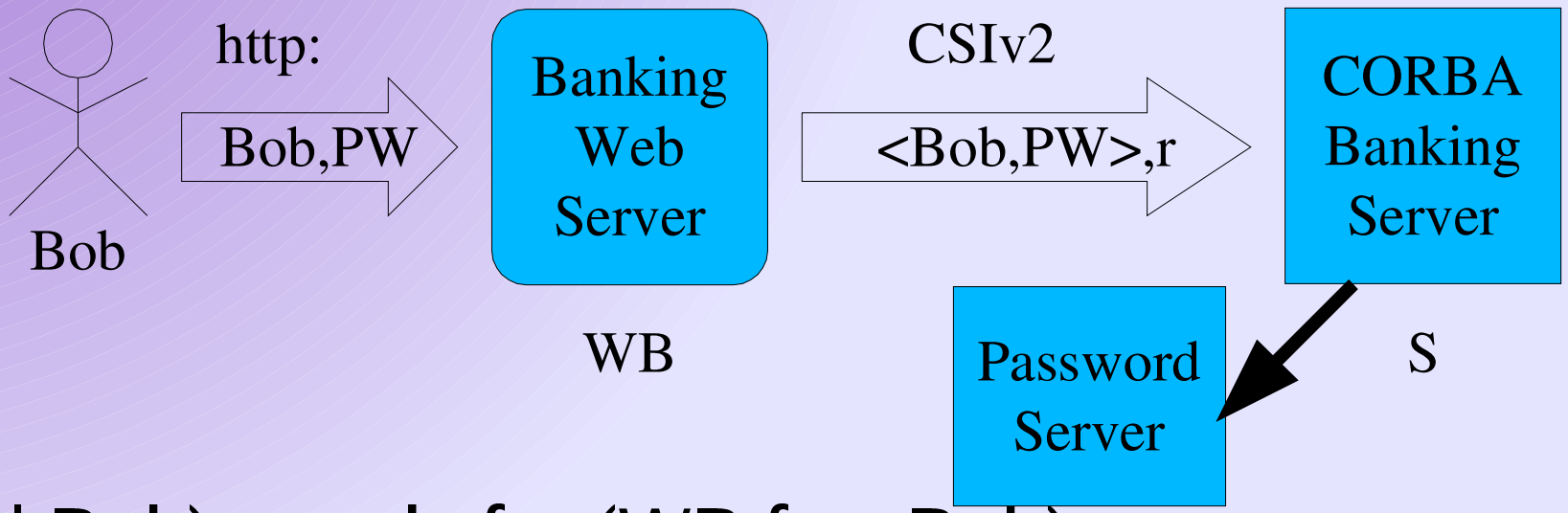


(WB | Bob) speaksfor (WB | Bob) /\ (S | Bob)

(WB | Bob) /\ (S | Bob) = (WB for$_S$ Bob)

(WB | Bob) speaksfor (WB for$_S$ Bob)

Example

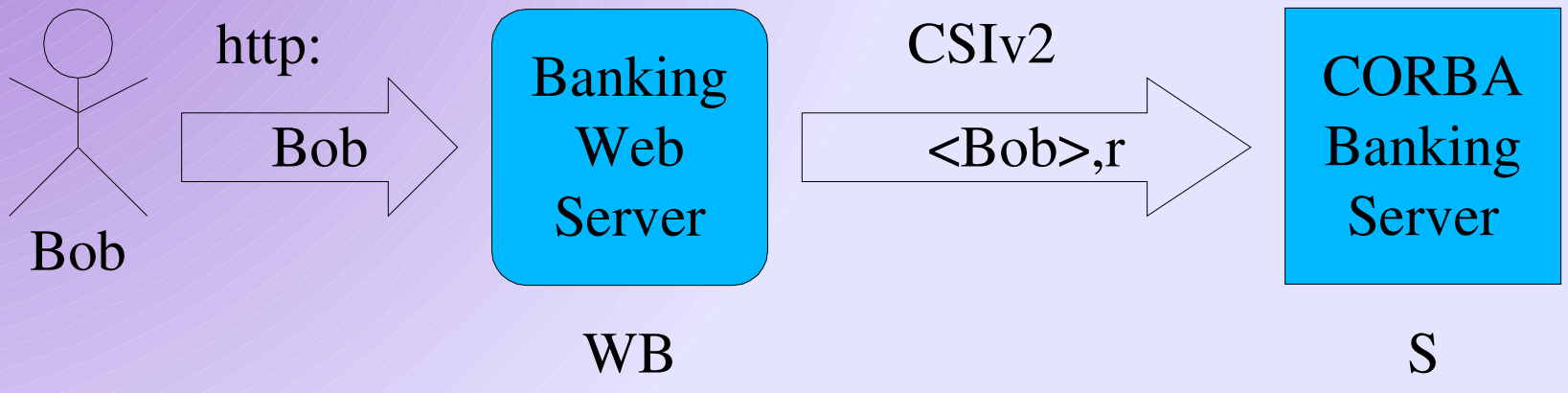


(WB | Bob) speaksfor (WB for$_S$ Bob)$_{PS}$

WB says Bob says r = (WB | Bob) says r

(WB for$_S$ Bob) says r. **ACCESS GRANTED!**

Example



WB says Bob says r

It is not the case that WB $\leq$ S

(WB | Bob) $\leq$ (WB for$_S$ Bob) does not hold!

ACCESS DENIED!

Conclusions

- We've formalized a small subset of CSIv2.
- We've formalized Userid Password and the rules in which must be in place for it to work.
- Formal Methods are cool!
  - They give the basis for assurance and verification.
- Formal Methods are needed at the design phase.
  - CSIv2 was developed with this calculus in hand.