Sheherazade's *Tale of the Three Judges*

An example of stepwise development of security protocols

Annabelle McIver

Carroll Morgan

The three-judges protocol

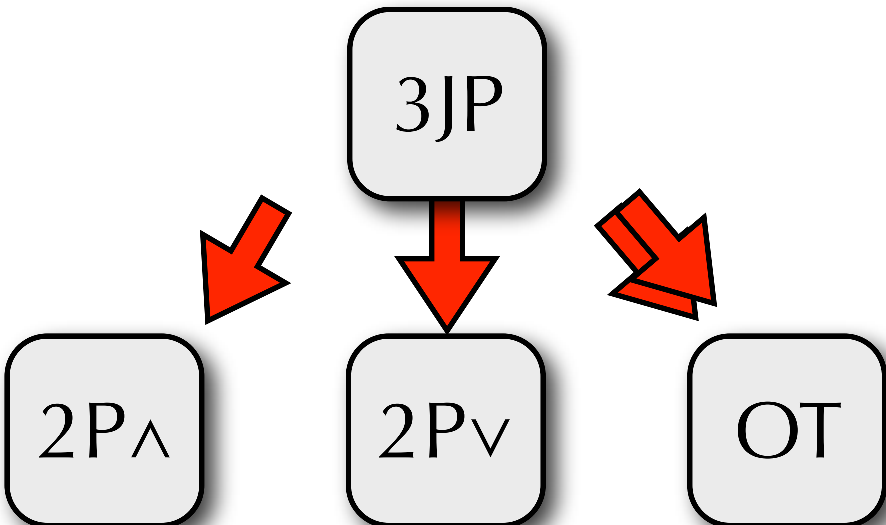
Three judge ‘bots communicate over the internet to reach a verdict by majority: but no judge’s individual decision is to be revealed.

3JP

The three-judges protocol

The three-judges protocol

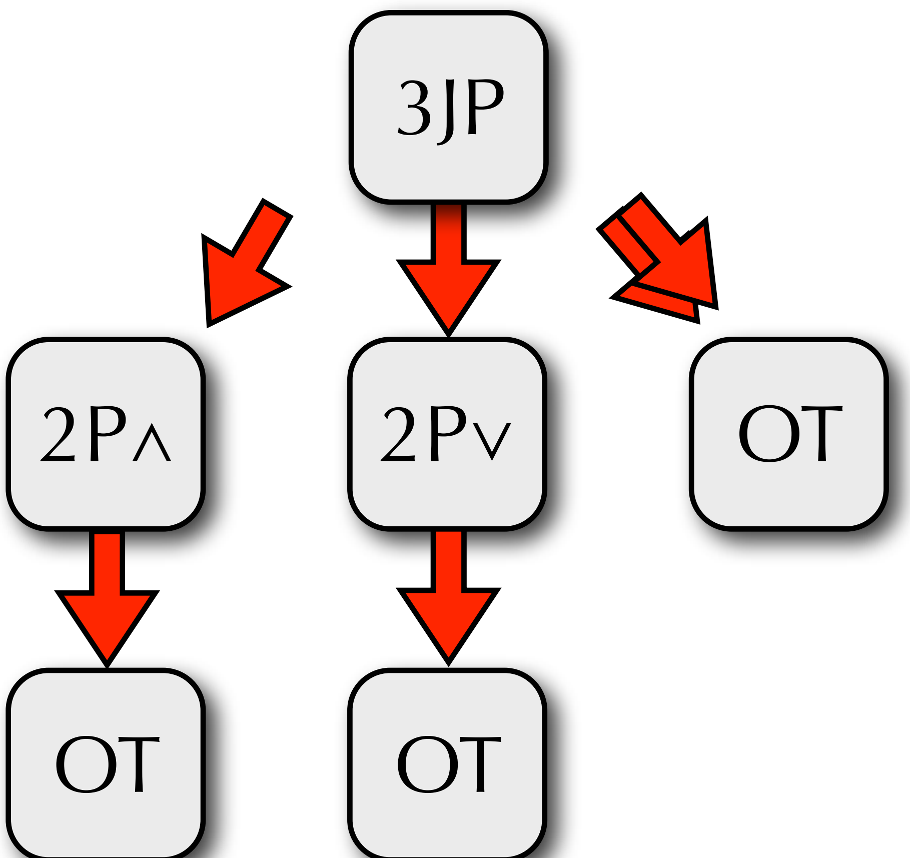
Three judge 'bots communicate over the internet to reach a verdict by majority: but no judge's individual decision is to be revealed.



The three-judges protocol
Two-party conjunction
Two-party disjunction
Oblivious transfer
Oblivious transfer

The three-judges protocol

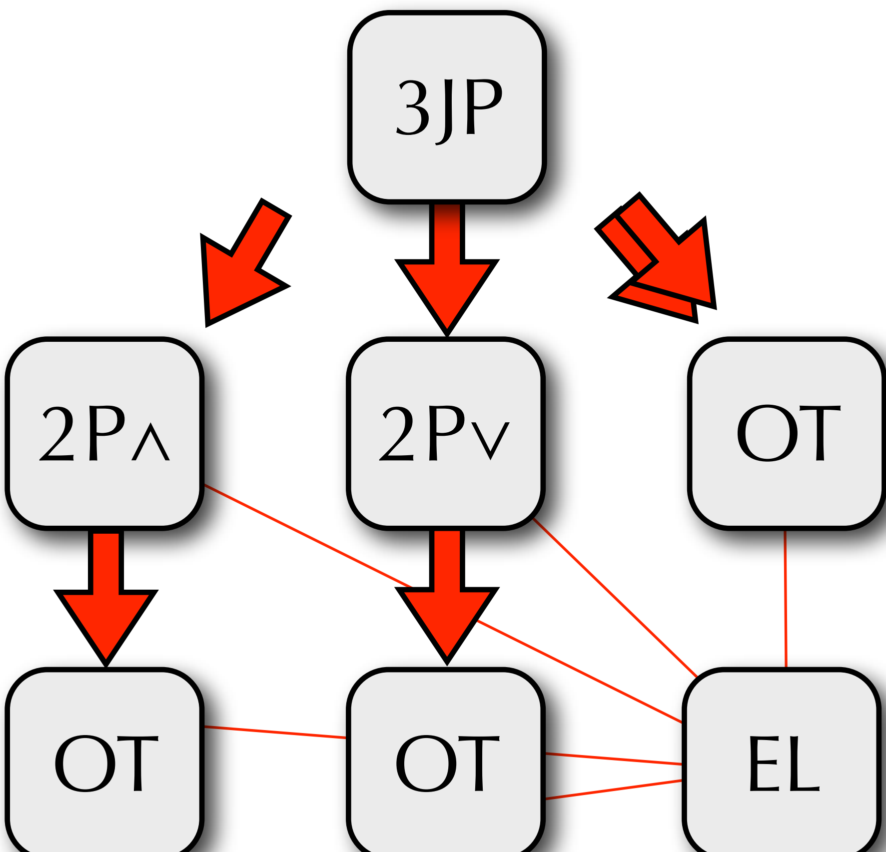
Three judge ‘bots communicate over the internet to reach a verdict by majority: but no judge’s individual decision is to be revealed.



The three-judges protocol
Two-party conjunction
Two-party disjunction
Oblivious transfer
Oblivious transfer
Oblivious transfer
Oblivious transfer

The three-judges protocol

Three judge 'bots communicate over the internet to reach a verdict by majority: but no judge's individual decision is to be revealed.



The three-judges protocol
Two-party conjunction
Two-party disjunction
Oblivious transfer
Oblivious transfer
Oblivious transfer
Oblivious transfer

Encryption lemma

Two-party conjunction

$$\begin{array}{l} \mathbf{hid}\ \mathrm{Bool}\ b, c \\ \quad \mathbf{reveal}\ b \wedge c \end{array}$$

Two-party conjunction

The Lovers' Protocol

specification

$$\textbf{hid}\ \mathrm{Bool}\ b, c$$
$$\textbf{reveal}\ b \wedge c$$

classical refinement

Agent B reveals a Boolean b_1 and Agent C reveals a Boolean c_0 such that the exclusive-or $b_1 \oplus c_0$ is the conjunction $b \wedge c$.

plus *secure refinement*

But neither b nor c is itself revealed in the process.

Two-party conjunction

$$\begin{aligned}&\mathbf{hid}\ \mathrm{Bool}\ b, c\\&\quad \mathbf{skip}\\&\quad \mathbf{reveal}\ b \wedge c\end{aligned}$$

Two-party conjunction

$$\textbf{hid}\ \mathrm{Bool}\ b, c$$

$$[\![\quad \textbf{hid}\ \mathrm{Bool}\ b_1, b_2$$

Encryption Lemma

$$b_1 \oplus b_2 := b$$

$$\textbf{reveal}\ b_1$$

$$]\!]$$

$$\textbf{reveal}\ b \wedge c$$

A common component of many protocols of this sort, including the Dining Cryptographers (Chaum) and the Oblivious Transfer (Rabin/Rivest).

Two-party conjunction

$$
\begin{array}{ll}
|[& \textbf{hid } \mathrm{Bool}\ b_1, b_2 \\
 & b_1 \oplus b_2 := b \\
 & \textbf{reveal } b_1 \\
]| &
\end{array}
$$

$$\textbf{reveal } b \wedge c$$

Two-party conjunction

$$
\begin{array}{ll}
|[& \mathbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
 & b_1 \oplus b_2 := b \\
 & \mathbf{reveal}\ b_1 \\
 & \mathbf{reveal}\ b \wedge c \\
]| &
\end{array}
$$

Two-party conjunction

$$
\begin{array}{ll}
|[& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
 & b_1 \oplus b_2 := b \\
 & \textbf{reveal}\ b_1 \\
 & \textbf{reveal}\ b \lhd c \rhd \textbf{false} \\
]| &
\end{array}
$$

IF

ELSE

Two-party conjunction

$$
\begin{array}{ll}
[\![& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
& b_1 \oplus b_2 := b \\
& \textbf{reveal}\ b_1 \\
& \textbf{reveal}\ (b_1 \oplus b_2) \lhd c \rhd (b_1 \oplus b_1) \\
]\!] &
\end{array}
$$

Two-party conjunction

$$
\begin{array}{ll}
|[& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
 & b_1 \oplus b_2 := b \\
 & \textbf{reveal}\ b_1 \\
 & \textbf{reveal}\ b_1 \oplus (b_2 \lhd c \rhd b_1) \\
]| &
\end{array}
$$

Two-party conjunction

$$
\begin{array}{ll}
[\![& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
& b_1 \oplus b_2 := b \\
& \textbf{reveal}\ b_1 \\
& \textbf{reveal}\ b_2 \lhd c \rhd b_1 \\
]\!] &
\end{array}
$$

Two-party conjunction

$$\left|\left[\right.\right.$$

$$\textbf{hid}\ \mathrm{Bool}\ b_1, b_2$$

$$b_1 \oplus b_2 := b$$

$$\textbf{reveal}\ b_1$$

$$\textbf{reveal}\ b_2 \lhd c \rhd b_1$$

replace by subprotocol

$$\left.\left.\right]\right|$$

Two-party conjunction

$$
\begin{aligned}
|[\quad & \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
& b_1 \oplus b_2 := b \\
& \textbf{reveal}\ b_1
\end{aligned}
$$

$$c_0 := b_2 \lhd c \rhd b_1$$ *a subprotocol*

$$]|$$

Two-party conjunction

$$
\begin{array}{l}
[\![\quad \mathbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
\quad b_1 \oplus b_2 := b \\
\quad \mathbf{reveal}\ b_1 \\
\quad [\![\quad \mathbf{hid}\ \mathrm{Bool}\ c_0 \\
\quad\quad c_0 := b_2 \lhd c \rhd b_1 \\
\quad\quad \mathbf{reveal}\ c_0 \\
\quad]\!] \\
]\!]
\end{array}
$$

a subprotocol

Two-party conjunction

$$
\begin{array}{ll}
[\![& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
 & \textbf{hid}\ \mathrm{Bool}\ c_0 \\
 & \\
 & b_1 \oplus b_2 := b \\
 & \textbf{reveal}\ b_1 \\
 & c_0 := b_2 \lhd c \rhd b_1 \\
 & \textbf{reveal}\ c_0 \\
]\!] &
\end{array}
$$

Two-party conjunction

$$
\begin{array}{ll}
[\![& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 \qquad \textit{B holds these} \\
& \textbf{hid}\ \mathrm{Bool}\ c_0 \\
& \\
& b_1 \oplus b_2 := b \\
& \textbf{reveal}\ b_1 \\
& c_0 := b_2 \lhd c \rhd b_1 \\
& \textbf{reveal}\ c_0 \\
]\!] &
\end{array}
$$

Two-party conjunction

$$
\begin{array}{ll}
|[\quad \mathbf{hid}\ \mathrm{Bool}\ b_1, b_2 & \textit{B}\text{ holds these} \\
\quad\ \ \mathbf{hid}\ \mathrm{Bool}\ c_0 & \textit{C}\text{ holds this} \\
\\
\quad\ \ b_1 \oplus b_2 := b & \\
\quad\ \ \mathbf{reveal}\ b_1 & \\
\quad\ \ c_0 := b_2 \lhd c \rhd b_1 & \\
\quad\ \ \mathbf{reveal}\ c_0 & \\
]| &
\end{array}
$$

Two-party conjunction

$$
\begin{array}{ll}
|[\quad \mathbf{hid}\ \mathrm{Bool}\ b_1, b_2 & \textit{B}\ \text{holds these} \\
\quad \mathbf{hid}\ \mathrm{Bool}\ c_0 & \textit{C}\ \text{holds this} \\
& \\
\left.\begin{array}{l} b_1 \oplus b_2 := b \\ \mathbf{reveal}\ b_1 \end{array}\right\} & \textit{B}\ \text{does these} \\
\quad c_0 := b_2 \lhd c \rhd b_1 & \\
\quad \mathbf{reveal}\ c_0 & \\
]| &
\end{array}
$$

Two-party conjunction

$$
\begin{array}{lll}
[\![& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 & \textit{B}\ \text{holds these} \\
 & \textbf{hid}\ \mathrm{Bool}\ c_0 & \textit{C}\ \text{holds this} \\
 & & \\
 & b_1 \oplus b_2 := b & \\
 & \textbf{reveal}\ b_1 & \left.\right\}\ \textit{B}\ \text{does these} \\
 & c_0 := b_2 \lhd c \rhd b_1 & \\
 & \textbf{reveal}\ c_0 & \textit{C}\ \text{does this} \\
]\!] & &
\end{array}
$$

Two-party conjunction

$$\begin{array}{lll}
|[& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 & B \text{ holds these} \\
 & \textbf{hid}\ \mathrm{Bool}\ c_0 & C \text{ holds this} \\
 & & \\
 & b_1 \oplus b_2 := b & \\
 & \textbf{reveal}\ b_1 & \left.\right\} B \text{ does these} \\
 & c_0 := b_2 \lhd c \rhd b_1 & \textit{Oblivious Transfer} \\
 & \textbf{reveal}\ c_0 & C \text{ does this} \\
]| & &
\end{array}$$

Due to *Rabin*; we use *Rivest's* formulation. An algebraic derivation of it is given in The Shadow Knows. Carroll Morgan. *Sci. Comp. Prog.* 2009, to appear.

Two-party conjunction:

$$
\begin{array}{ll}
[\![\quad \mathbf{hid}\ \mathrm{Bool}\ b_1, b_2 & \textit{B} \text{ holds these} \\
\quad \mathbf{hid}\ \mathrm{Bool}\ c_0 & \textit{C} \text{ holds this} \\
& \\
\quad b_1 \oplus b_2 := b & \\
\quad \mathbf{reveal}\ b_1 & \left.\right\} \textit{B} \text{ does these} \\
\quad c_0 := b_2 \lhd c \rhd b_1 & \textit{Oblivious Transfer} \\
\quad \mathbf{reveal}\ c_0 & \textit{C} \text{ does this} \\
]\!] &
\end{array}
$$

Two-party conjunction: externally viewed

$$
\begin{array}{ll}
[\![& \\
\quad \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 & \textit{B}\ \text{holds these} \\
\quad \textbf{hid}\ \mathrm{Bool}\ c_0 & \textit{C}\ \text{holds this} \\
& \\
\quad \left.\begin{array}{l} b_1 \oplus b_2 := b \\ \textbf{reveal}\ b_1 \end{array}\right\} & \textit{B}\ \text{does these} \\
\quad c_0 := b_2 \lhd c \rhd b_1 & \textit{Oblivious Transfer} \\
\quad \textbf{reveal}\ c_0 & \textit{C}\ \text{does this} \\
]\!] &
\end{array}
$$

Two-party conjunction:

Two-party conjunction: as seen by *B*

Variables' names usually indicate where they are located, i.e. which agent "owns" them. It's an informal convention in this talk. It can of course be made precise with annotations, but we don't bother.

On the other hand, we do bother for visibility attributes: an agent sometimes can see variables owned by others, and sometimes cannot.

Two-party conjunction: as seen by B

$$\textbf{vis}\ \text{Bool}\ b$$
$$\textbf{hid}\ \text{Bool}\ c$$

$$|[\quad \textbf{vis}\ \text{Bool}\ b_1, b_2$$
$$\textbf{hid}\ \text{Bool}\ c_0$$

$$b_1 \oplus b_2 := b; \qquad \textbf{reveal}\ b_1$$
$$c_0 := b_2 \lhd c \rhd b_1; \quad \textbf{reveal}\ c_0$$
$$]|$$

Two-party conjunction:

$$\textbf{hid}\ \mathrm{Bool}\ b$$
$$\textbf{vis}\ \mathrm{Bool}\ c$$

$$\begin{array}{lll} |[& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 & \\ & \textbf{vis}\ \mathrm{Bool}\ c_0 & \\ & & \\ & b_1 \oplus b_2 := b; & \textbf{reveal}\ b_1 \\ & c_0 := b_2 \lhd c \rhd b_1; & \textbf{reveal}\ c_0 \\ ]| & & \end{array}$$

Two-party conjunction: as seen by C

$$\textbf{hid}\ \mathrm{Bool}\ b$$
$$\textbf{vis}\ \mathrm{Bool}\ c$$

$$\begin{array}{ll}
[\![& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
 & \textbf{vis}\ \mathrm{Bool}\ c_0 \\
 & \\
 & b_1 \oplus b_2 := b; \qquad\qquad \textbf{reveal}\ b_1 \\
 & c_0 := b_2 \lhd c \rhd b_1; \quad \textbf{reveal}\ c_0 \\
]\!] &
\end{array}$$

Two-party conjunction: multiple views

$\mathbf{vis}_B$ Bool b
$\mathbf{vis}_C$ Bool c

specification

$\mathbf{reveal}\ b \wedge c$

$\llbracket$ $\mathbf{vis}_B$ Bool b_1, b_2

implementation

$\mathbf{vis}_C$ Bool c_0

$b_1 \oplus b_2 := b;$ $\quad$ $\mathbf{reveal}\ b_1$
$c_0 := b_2 \lhd c \rhd b_1;$ $\quad$ $\mathbf{reveal}\ c_0$
$\rrbracket$

The tale of the Three Judges

specification

$$\textbf{vis}_A\ \{0,1\}\ a$$
$$\textbf{vis}_B\ \{0,1\}\ b$$
$$\textbf{vis}_C\ \{0,1\}\ c$$

$$\textbf{reveal}\ (a+b+c\geq 2)$$

Reveal the majority verdict.

secure refinement

Do not reveal individual verdicts to anyone.

The tale of the Three Judges

$$\mathbf{reveal}\ (a + b + c \geq 2)$$

The tale of the Three Judges

$$\textbf{reveal}\ (a + b + c \geq 2)$$

$$\begin{aligned} & a{+}b{+}c \geq 2 \\ \equiv \quad & a \wedge (b \vee c) \quad \vee \quad b \wedge c \end{aligned}$$

The tale of the Three Judges

$$\textbf{reveal}\ (a + b + c \geq 2)$$

$$\begin{array}{rl} & a{+}b{+}c \geq 2 \\ \equiv & a \wedge (b \vee c) \quad \vee \quad b \wedge c \\ \equiv & b \vee c \quad \lhd\, a \,\rhd \quad b \wedge c \end{array}$$

The tale of the Three Judges

$$\textbf{reveal}\ (a + b + c \geq 2)$$

$$\begin{aligned}
& a{+}b{+}c \geq 2 \\
\equiv \quad & a \wedge (b \vee c) \quad \vee \quad b \wedge c \\
\equiv \quad & \boxed{b \vee c \quad \triangleleft\, a\, \triangleright \quad b \wedge c}
\end{aligned}$$

A shortcut.

The tale of the Three Judges: First attempt

$$
\begin{array}{l}
\textbf{if } a \\
\quad \textbf{then } \ \textbf{reveal } (a+b+c \geq 2) \\
\quad \textbf{else } \ \ \textbf{reveal } (a+b+c \geq 2) \\
\textbf{fi}
\end{array}
$$

But watch out... We will return to this.

The tale of the Three Judges: First attempt

$$\begin{array}{l}\textbf{if } a \\ \quad \textbf{then} \;\; \textbf{reveal } b \vee c \\ \quad \textbf{else} \;\; \textbf{reveal } b \wedge c \\ \textbf{fi}\end{array}$$

The tale of the Three Judges: First attempt

$$
\begin{array}{l}
\textbf{if } a \\
\quad \textbf{then} \;\; \textbf{reveal } b \vee c \qquad \neg(\neg b \wedge \neg c) \\
\quad \textbf{else} \;\;\; \textbf{reveal } b \wedge c \\
\textbf{fi}
\end{array}
$$

The tale of the Three Judges: First attempt

$$
\begin{array}{l}
\textbf{if } a \\
\quad \textbf{then} \quad \textbf{reveal } b \vee c \\
\quad \textbf{else} \quad \textbf{reveal } b \wedge c \\
\textbf{fi}
\end{array}
$$

The tale of the Three Judges: First attempt

$$
\begin{array}{l}
\textbf{if } a \\
\quad \textbf{then } \textbf{reveal } b \vee c \\
\quad \textbf{else} \\
\\
\\
\\
\textbf{fi}
\end{array}
$$

The tale of the Three Judges: First attempt

$$
\begin{array}{l}
\textbf{if } a \\
\quad \textbf{then} \;\; \textbf{reveal } b \vee c \\
\quad \textbf{else} \;\;\; b_1 \oplus b_2 := b \\
\qquad\qquad\;\; c_0 := b_2 \triangleleft c \triangleright b_1 \\
\\
\textbf{fi}
\end{array}
$$

The tale of the Three Judges: First attempt

$$
\begin{array}{lll}
\textbf{if } a & & \\
& \textbf{then} & \textbf{reveal } b \vee c \\
& \textbf{else} & b_1 \oplus b_2 := b \\
& & c_0 := b_2 \lhd c \rhd b_1 \\
& & a_b := b_1 \\
& & a_c := c_0 \\
\textbf{fi} & &
\end{array}
$$

The tale of the Three Judges: First attempt

$\mathbf{vis}_A\ a_b, a_c$
$\mathbf{vis}_B\ b_1, b_2$
$\mathbf{vis}_C\ c_0$

$$
\begin{array}{l}
\mathbf{if}\ a \\
\quad \mathbf{then}\ \ \mathbf{reveal}\ b \vee c \\
\quad \mathbf{else}\ \ \ b_1 \oplus b_2 := b \\
\qquad\qquad\ c_0 := b_2 \lhd c \rhd b_1 \\
\qquad\qquad\ a_b := b_1 \\
\qquad\qquad\ a_c := c_0 \\
\qquad\qquad\ \mathbf{reveal}\ a_b \oplus a_c \\
\mathbf{fi}
\end{array}
$$

The tale of the Three Judges: First attempt

$$\textbf{vis}_A\ a_b, a_c$$
$$\textbf{vis}_B\ b_1, b_2$$
$$\textbf{vis}_C\ c_0$$

$$\textbf{if}\ a$$
$$\textbf{then}\quad b_1 \equiv b_2 := b$$
$$c_0 := b_2 \lhd c \rhd b_1$$
$$a_b := b_1$$
$$a_c := c_0$$
$$\textbf{reveal}\ a_b \oplus a_c$$

The tale of the Three Judges: First attempt

$\textbf{vis}_A \; a_b, a_c$
$\textbf{vis}_B \; b_1, b_2$
$\textbf{vis}_C \; c_0$

$$
\begin{array}{ll}
\textbf{if } a & \\
\quad \textbf{then} & b_1 \equiv b_2 := b \\
 & c_0 := b_2 \lhd c \rhd b_1 \\
 & a_b := b_1 \\
 & a_c := c_0 \\
 & \textbf{reveal } a_b \oplus a_c
\end{array}
$$

Oops! Agent *B* learns *a* by noting which protocol it is asked to follow.

The tale of the Three Judges: First attempt

$$
\begin{array}{lll}
 & prog & \\
\neq & \textbf{if } a & \\
 & \quad \textbf{then} & prog \\
 & \quad \textbf{else} & prog \\
 & \textbf{fi} &
\end{array}
$$

Because the two right-hand instances of *prog* can be manipulated independently, the Shadow semantics does not allow this equality.

The tale of the Three Judges: First attempt

$$\textbf{reveal}\ (a+b+c \geq 2)$$

$$\neq \quad \begin{array}{l} \textbf{if}\ a \\ \quad \textbf{then}\ \ \textbf{reveal}\ (a+b+c \geq 2) \\ \quad \textbf{else}\ \ \ \textbf{reveal}\ (a+b+c \geq 2) \\ \textbf{fi} \end{array}$$

Not allowed.

The tale of the Three Judges:

$$\mathbf{reveal}\ (a + b + c \geq 2)$$

$$\stackrel{?}{=} \quad \text{“}\mathit{Get\ both}\ b \vee c\ \mathit{and}\ b \wedge c\text{”}$$

$$\mathbf{reveal}\ (b \vee c) \lhd a \rhd (b \wedge c)$$

This way, Agent *B* cannot tell which of the propositions Agent *A* actually wants.

The tale of the Three Judges: Second attempt

$$\textbf{reveal } (a + b + c \geq 2)$$

$$\stackrel{?}{=} \quad \text{“}\textit{Get both } b \vee c \textit{ and } b \wedge c\text{”}$$

$$\textbf{reveal } (b \vee c) \lhd a \rhd (b \wedge c)$$

This way, Agent B cannot tell which of the propositions Agent A actually wants.

The tale of the Three Judges: Second attempt

$$\textbf{reveal } (a + b + c \geq 2)$$

$$\neq \quad \text{“}\textit{Get both } b \vee c \textit{ and } b \wedge c\text{”}$$

$$\textbf{reveal } (b \vee c) \lhd a \rhd (b \wedge c)$$

Oops! Agent A learns both $b \vee c$ and $b \wedge c$.

The tale of the Three Judges:

$$\textbf{reveal } (a + b + c \geq 2)$$

$$\begin{aligned} = \quad & b_\vee \oplus c_\vee := b \vee c \\ & b_\wedge \oplus c_\wedge := b \wedge c \\ & a_b := (b_\vee \lhd a \rhd b_\wedge) \\ & a_c := (c_\vee \lhd a \rhd c_\wedge) \\ & \textbf{reveal } a_b \oplus a_c \end{aligned}$$

$$\begin{aligned} & \textbf{vis}_A \; a_b, a_c \\ & \textbf{vis}_B \; b_\wedge, b_\vee \\ & \textbf{vis}_C \; c_\wedge, c_\vee \end{aligned}$$

The tale of the Three Judges: Correct refinement

$$\textbf{reveal}\ (a+b+c \geq 2)$$

$$
\begin{aligned}
= \quad & b_\vee \oplus c_\vee := b \vee c \\
& b_\wedge \oplus c_\wedge := b \wedge c \\
& a_b := (b_\vee \triangleleft a \triangleright b_\wedge) \\
& a_c := (c_\vee \triangleleft a \triangleright c_\wedge) \\
& \textbf{reveal}\ a_b \oplus a_c
\end{aligned}
$$

$\textbf{vis}_A\ a_b, a_c$
$\textbf{vis}_B\ b_\wedge, b_\vee$
$\textbf{vis}_C\ c_\wedge, c_\vee$

Variable's name indicates its owner;
its subscript indicates its purpose.

The tale of the Three Judges: Correct refinement

$$\textbf{reveal } (a+b+c \geq 2)$$

$$
\begin{aligned}
= \quad & b_\vee \oplus c_\vee := b \vee c \\
& b_\wedge \oplus c_\wedge := b \wedge c \\
& a_b := (b_\vee \lhd a \rhd b_\wedge) \\
& a_c := (c_\vee \lhd a \rhd c_\wedge) \\
& \textbf{reveal } a_b \oplus a_c
\end{aligned}
$$

$\textbf{vis}_A\ a_b, a_c$
$\textbf{vis}_B\ b_\wedge, b_\vee$
$\textbf{vis}_C\ c_\wedge, c_\vee$

None of Agents A,B,C learns anything about $b \vee c$ from this.

The tale of the Three Judges: Correct refinement

$$\textbf{reveal}\ (a + b + c \geq 2)$$

$$
\begin{aligned}
= \quad & b_\vee \oplus c_\vee := b \vee c \\
& b_\wedge \oplus c_\wedge := b \wedge c \\
& a_b := (b_\vee \lhd a \rhd b_\wedge) \\
& a_c := (c_\vee \lhd a \rhd c_\wedge) \\
& \textbf{reveal}\ a_b \oplus a_c
\end{aligned}
$$

$\textbf{vis}_A\ a_b, a_c$
$\textbf{vis}_B\ b_\wedge, b_\vee$
$\textbf{vis}_C\ c_\wedge, c_\vee$

None of Agents A,B,C learns anything about $b \wedge c$ from this.

The tale of the Three Judges: Correct refinement

$$\mathbf{reveal}\ (a+b+c \geq 2)$$

$$
\begin{aligned}
= \quad & b_\vee \oplus c_\vee := b \vee c \\
& b_\wedge \oplus c_\wedge := b \wedge c \\
& a_b := (b_\vee \lhd a \rhd b_\wedge) \\
& a_c := (c_\vee \lhd a \rhd c_\wedge) \\
& \mathbf{reveal}\ a_b \oplus a_c
\end{aligned}
$$

$\mathbf{vis}_A\ a_b, a_c$
$\mathbf{vis}_B\ b_\wedge, b_\vee$
$\mathbf{vis}_C\ c_\wedge, c_\vee$

Agent *A* learns nothing about "the other" *b*; and Agent *B* learns nothing about *a*.

The tale of the Three Judges: Correct refinement

$$\textbf{reveal } (a + b + c \geq 2)$$

$$= \quad b_\vee \oplus c_\vee := b \vee c$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

$$a_c := (c_\vee \lhd a \rhd c_\wedge)$$

$$\textbf{reveal } a_b \oplus a_c$$

$\textbf{vis}_A \ a_b, a_c$
$\textbf{vis}_B \ b_\wedge, b_\vee$
$\textbf{vis}_C \ c_\wedge, c_\vee$

Agent A learns nothing about "the other" c; and Agent C learns nothing about a.

The tale of the Three Judges: Correct refinement

$$\textbf{reveal}\ (a+b+c \geq 2)$$

$$
\begin{aligned}
= \quad & b_\vee \oplus c_\vee := b \vee c \\
& b_\wedge \oplus c_\wedge := b \wedge c \\
& a_b := (b_\vee \lhd a \rhd b_\wedge) \\
& a_c := (c_\vee \lhd a \rhd c_\wedge) \\
& \textbf{reveal}\ a_b \oplus a_c
\end{aligned}
$$

$$
\begin{aligned}
&\textbf{vis}_A\ a_b, a_c \\
&\textbf{vis}_B\ b_\wedge, b_\vee \\
&\textbf{vis}_C\ c_\wedge, c_\vee
\end{aligned}
$$

In spite of all that, the verdict (a+b+$c \geq 2$) is revealed at this point.

The tale of the Three Judges: Correct refinement

$$\textbf{reveal}\ (a+b+c \geq 2)$$

$$
\begin{aligned}
= \quad & b_\vee \oplus c_\vee := b \vee c \\
& b_\wedge \oplus c_\wedge := b \wedge c \\
& a_b := (b_\vee \lhd a \rhd b_\wedge) \\
& a_c := (c_\vee \lhd a \rhd c_\wedge) \\
& \textbf{reveal}\ a_b \oplus a_c
\end{aligned}
$$

$\textbf{vis}_A\ a_b, a_c$
$\textbf{vis}_B\ b_\wedge, b_\vee$
$\textbf{vis}_C\ c_\wedge, c_\vee$

In spite of all that, the verdict (a+b+$c \geq 2$) is revealed at this point.

The tale of the Three Judges: Correct refinement

$$\textbf{reveal}\ (a + b + c \geq 2)$$

$$
\begin{aligned}
= \quad & b_\vee \oplus c_\vee := b \vee c \\
& b_\wedge \oplus c_\wedge := b \wedge c \\
& a_b := (b_\vee \triangleleft a \triangleright b_\wedge) \\
& a_c := (c_\vee \triangleleft a \triangleright c_\wedge) \\
& \textbf{reveal}\ a_b \oplus a_c
\end{aligned}
$$

$$
\begin{aligned}
&\textbf{vis}_A\ a_b, a_c \\
&\textbf{vis}_B\ b_\wedge, b_\vee \\
&\textbf{vis}_C\ c_\wedge, c_\vee
\end{aligned}
$$

In spite of all that, the verdict (a+b+$c \geq 2$) is revealed at this point.

The tale of the Three Judges: Correct refinement

$$\textbf{reveal}\ (a+b+c \geq 2)$$

$$
\begin{aligned}
= \quad & (b_\vee \equiv b'_\vee) := b; \\
& c_\vee := (b'_\vee \lhd c \rhd b_\vee); \\
& (b_\wedge \oplus b'_\wedge) := b; \\
& c_\wedge := (b'_\wedge \lhd c \rhd b_\wedge); \\
& a_b := (b_\vee \lhd a \rhd b_\wedge); \\
& a_c := (c_\vee \lhd a \rhd c_\wedge); \\
& \textbf{reveal}\ a_b \oplus a_c
\end{aligned}
$$

The tale of the Three Judges: Correct refinement

$$\textbf{reveal } (a+b+c \geq 2)$$

$$
\begin{array}{ll}
= & (b_\vee \equiv b'_\vee) := b; \\
 & c_\vee := (b'_\vee \lhd c \rhd b_\vee); \\
 & (b_\wedge \oplus b'_\wedge) := b; \\
 & c_\wedge := (b'_\wedge \lhd c \rhd b_\wedge); \\
 & a_b := (b_\vee \lhd a \rhd b_\wedge); \\
 & a_c := (c_\vee \lhd a \rhd c_\wedge); \\
 & \textbf{reveal } a_b \oplus a_c
\end{array}
$$

The tale of the Three Judges: Correct refinement

$$\textbf{reveal } (a+b+c \geq 2)$$

$$
\begin{array}{rl}
= & (b_\vee \equiv b'_\vee) := b; \\
 & c_\vee := (b'_\vee \lhd c \rhd b_\vee); \\
 & (b_\wedge \oplus b'_\wedge) := b; \\
 & c_\wedge := (b'_\wedge \lhd c \rhd b_\wedge); \\
 & a_b := (b_\vee \lhd a \rhd b_\wedge); \\
 & a_c := (c_\vee \lhd a \rhd c_\wedge); \\
 & \textbf{reveal } a_b \oplus a_c
\end{array}
$$

The tale of the Three Judges: Correct refinement

$$\textbf{reveal}\ (a+b+c \geq 2)$$

$$
\begin{array}{ll}
= & (b_\vee \equiv b'_\vee) := b; \\
& c_\vee := \textit{Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.} \\
& (b_\wedge \oplus b'_\wedge) := b; \\
& c_\wedge := \textit{Lorem ipsum dolor}
\end{array}
$$

The tale of the Three Judges: Correct refinement

$$\textbf{reveal}\ (a + b + c \geq 2)$$

$=$ “Source-level proof”

$(b_\vee \equiv b'_\vee) := b;$
$c_\vee :=$ *Lorem ipsum dolor*
sit amet, consectetur
adipisicing elit, sed
do eiusmod tempor
incididunt ut labore et
dolore magna aliqua.

$(b_\wedge \oplus b'_\wedge) := b;$
$c_\wedge :=$ *Lorem ipsum dolor*
sit amet, consectetur
adipisicing elit, sed
do eiusmod tempor
incididunt ut labore et
dolore magna aliqua.

$a_b :=$ *Lorem ipsum dolor*
sit amet, consectetur
adipisicing elit, sed
do eiusmod tempor
incididunt ut labore et
dolore magna aliqua.

$a_c :=$ *Lorem ipsum dolor*
sit amet, consectetur
adipisicing elit, sed
do eiusmod tempor
incididunt ut labore et
dolore magna aliqua.

The tale of the Three Judges: Correct refinement

$$\textbf{reveal } (a+b+c \geq 2)$$

$=$ "***This***-level proof"

$$b_\vee \oplus c_\vee := b \vee c$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

$$a_c := (c_\vee \lhd a \rhd c_\wedge)$$

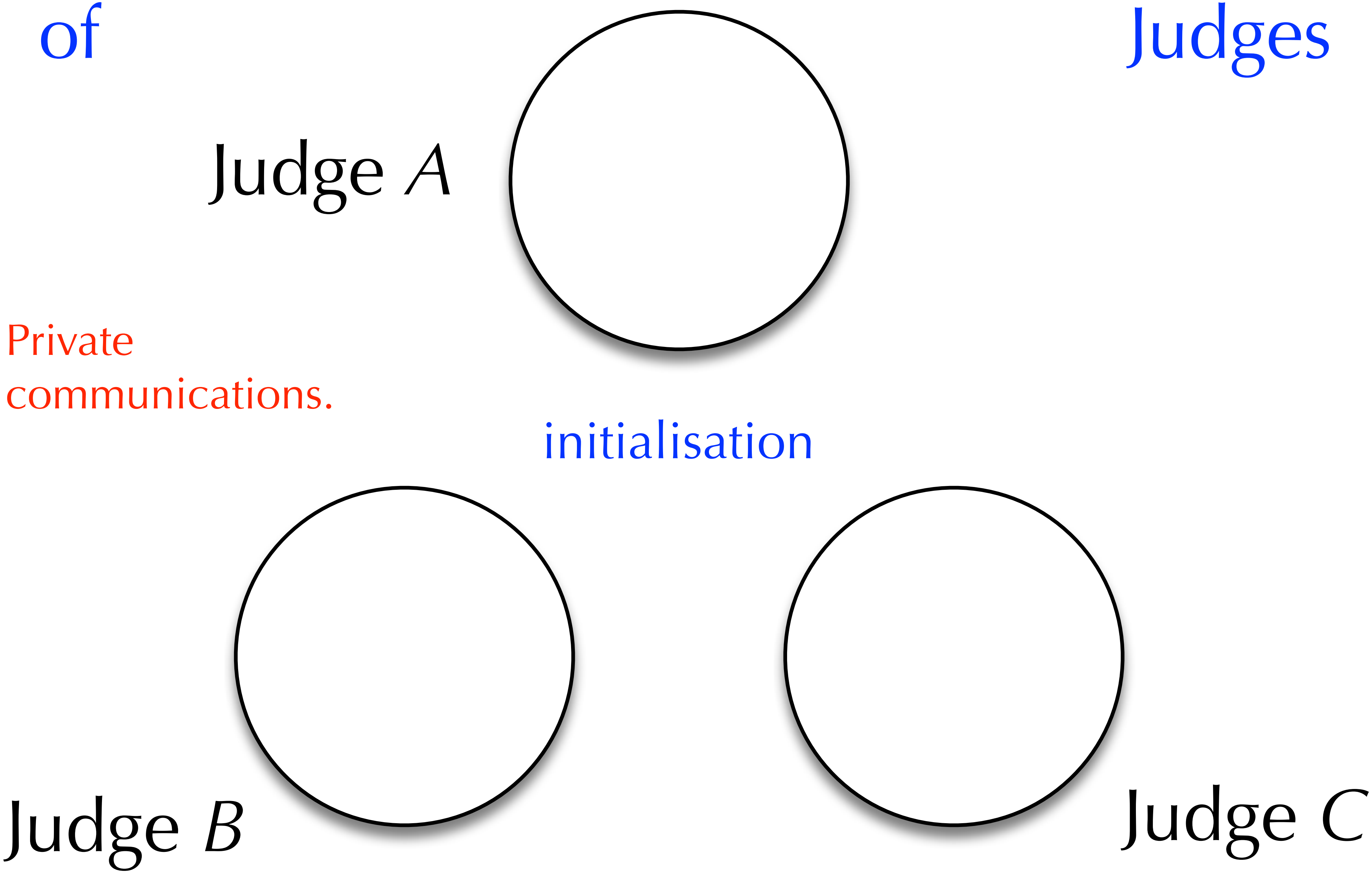
$$\textbf{reveal } a_b \oplus a_c$$

The subprotocols' proofs are (will be one day) *off-the-shelf*, and need not be repeated for specific applications.

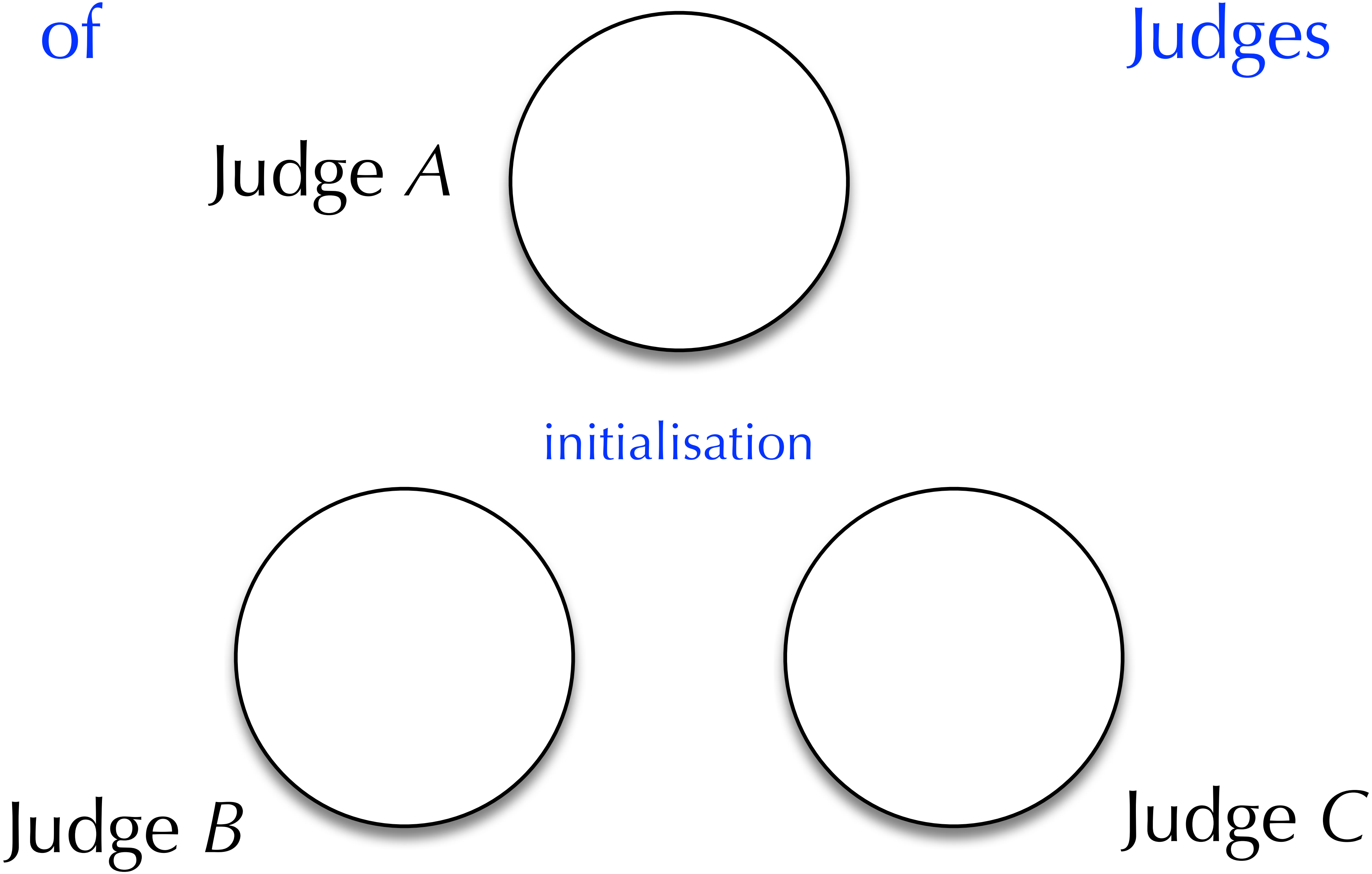
The tale of the Three Judges

The tale of the Three Judges

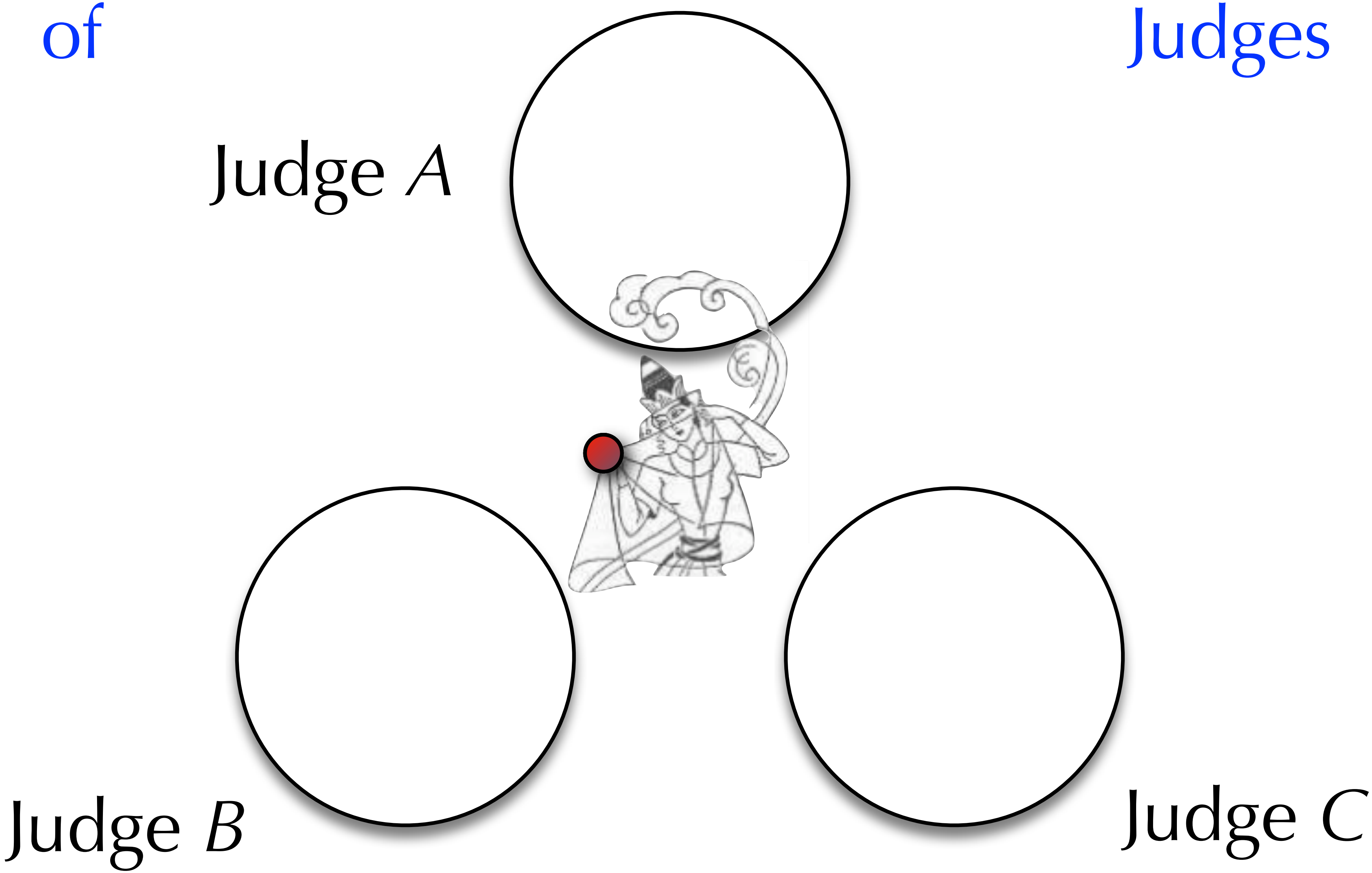
Private communications.



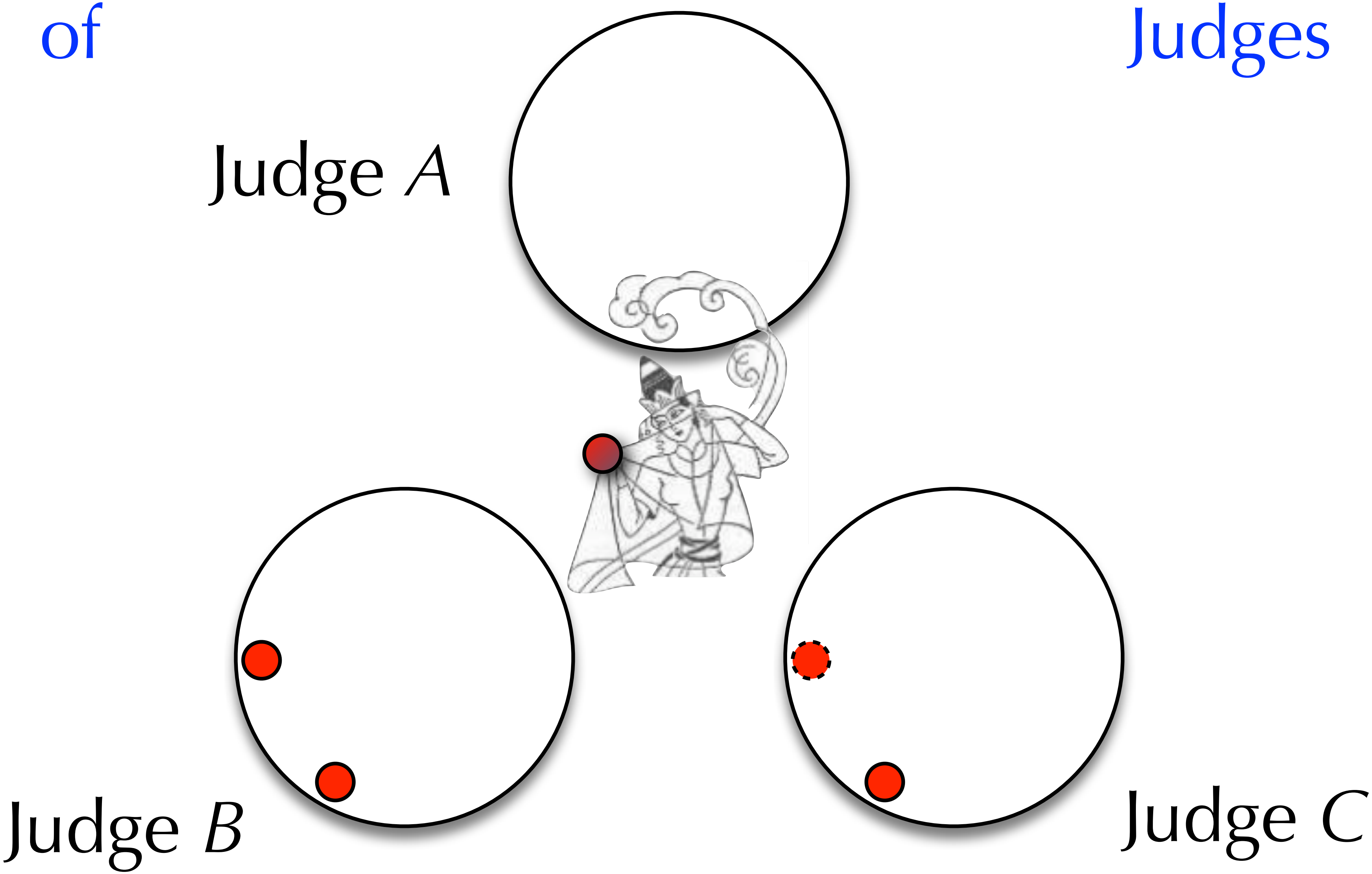
The tale of the Three Judges



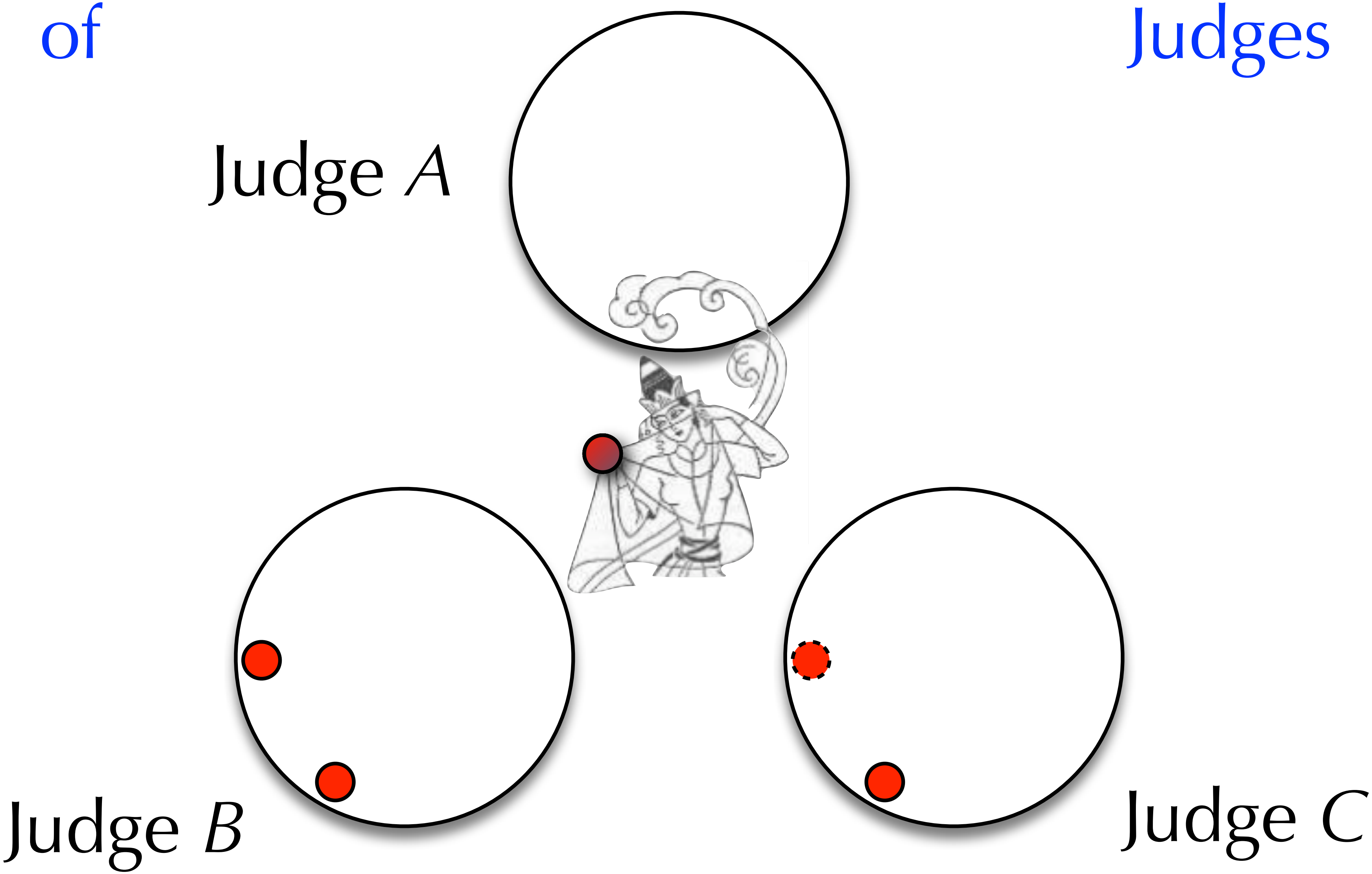
The tale of the Three Judges



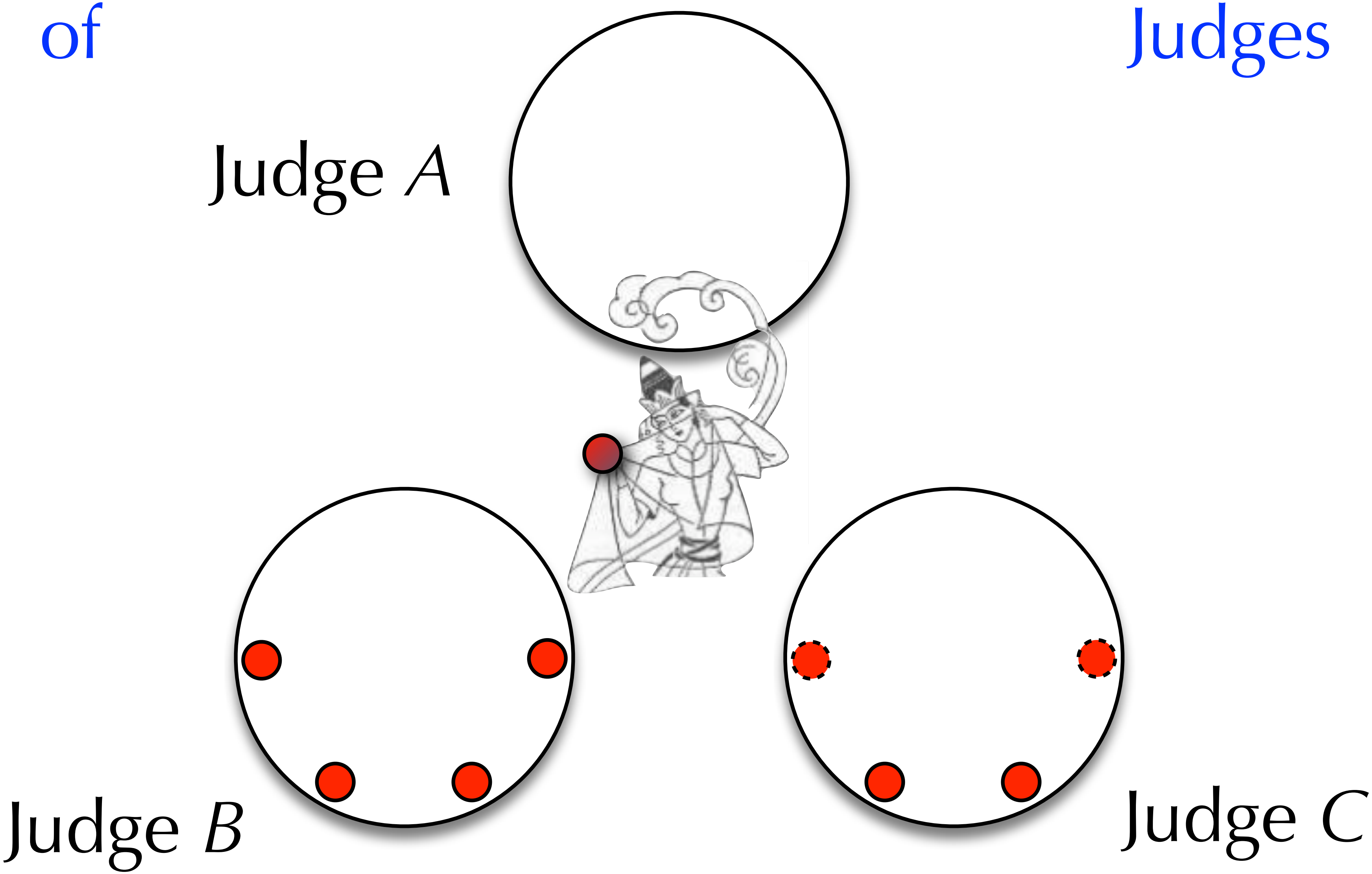
The tale of the Three Judges



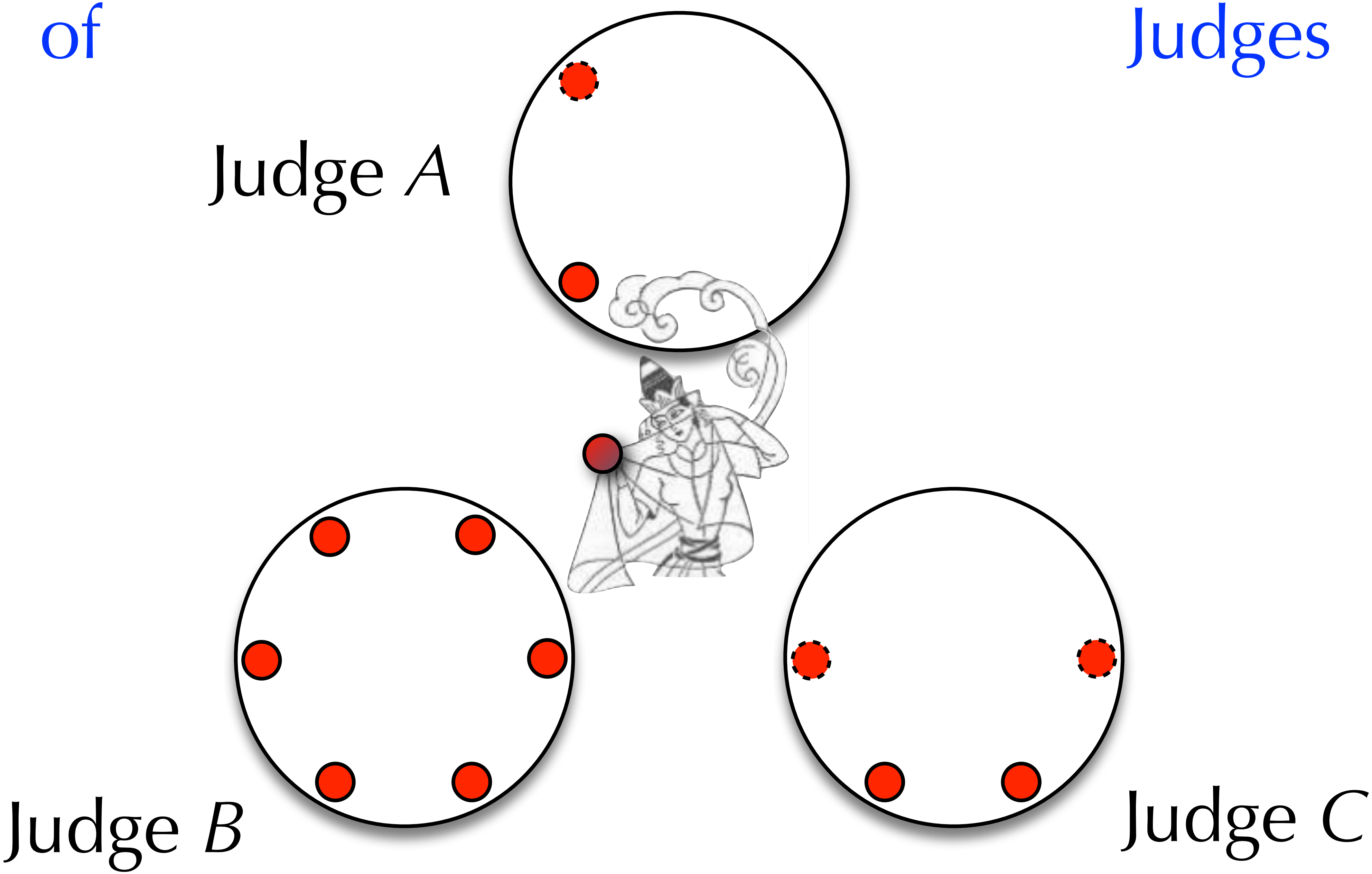
The tale of the Three Judges



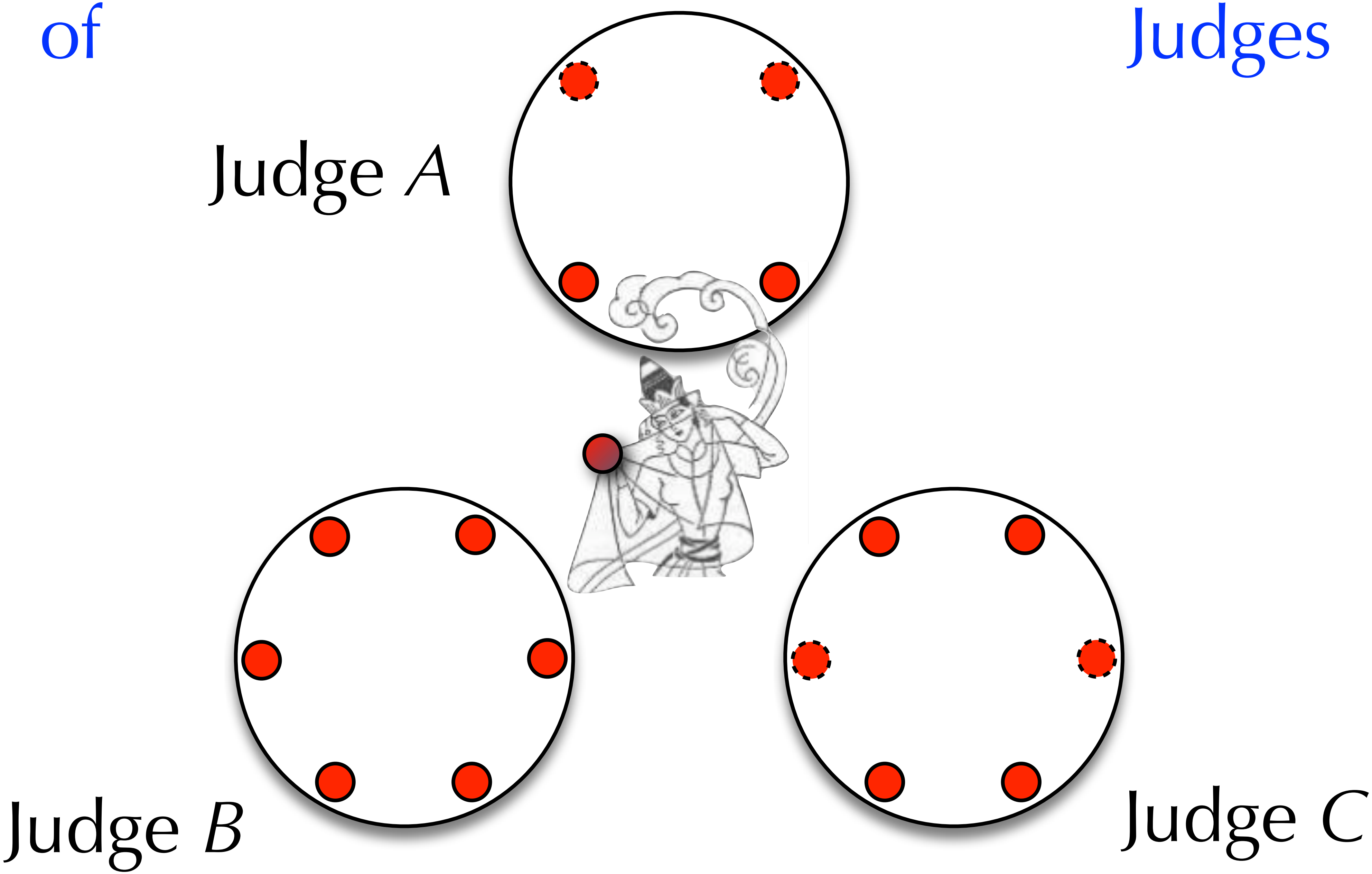
The tale of the Three Judges



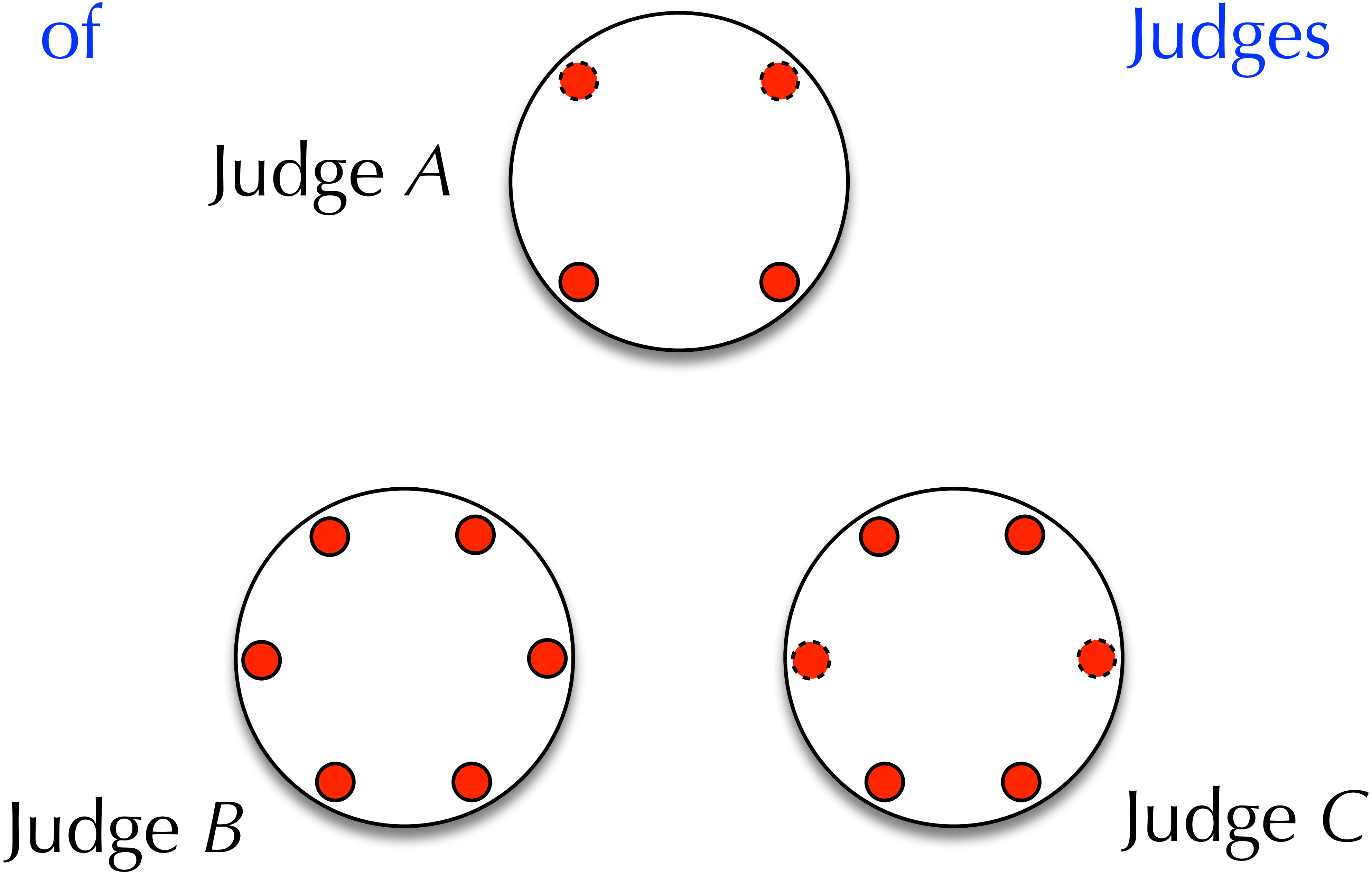
The tale of the Three Judges



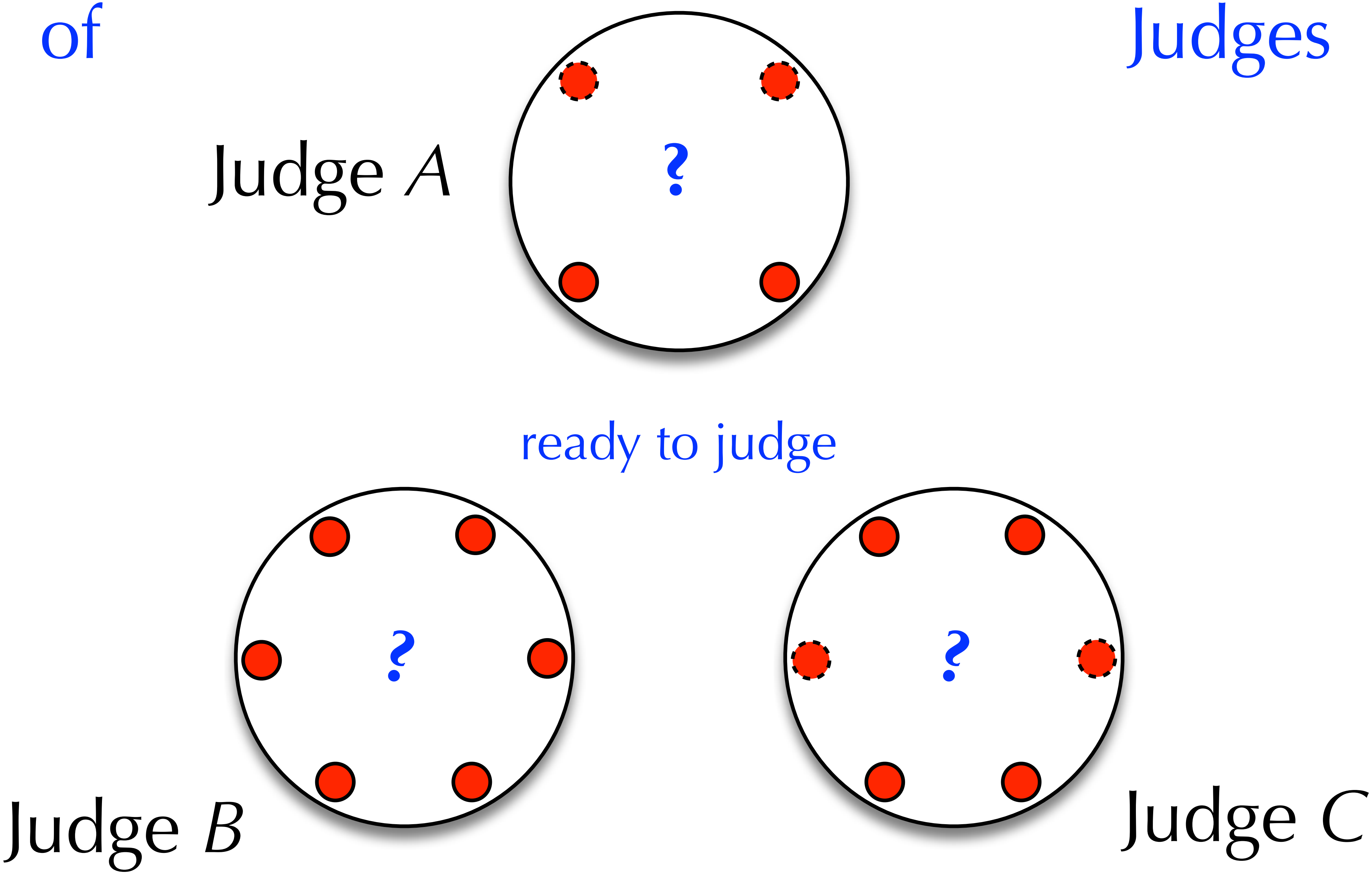
The tale of the Three Judges



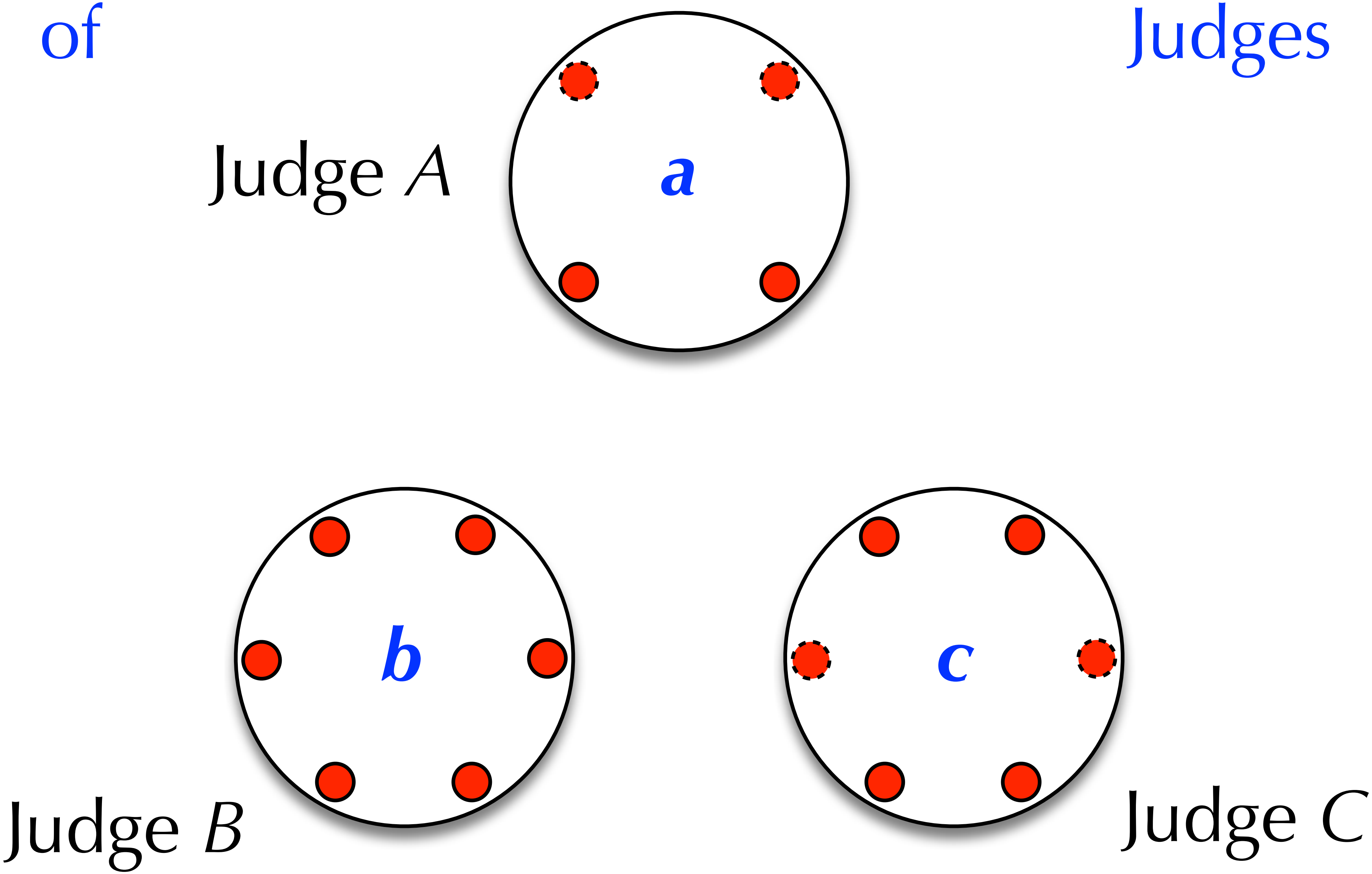
The tale of the Three Judges



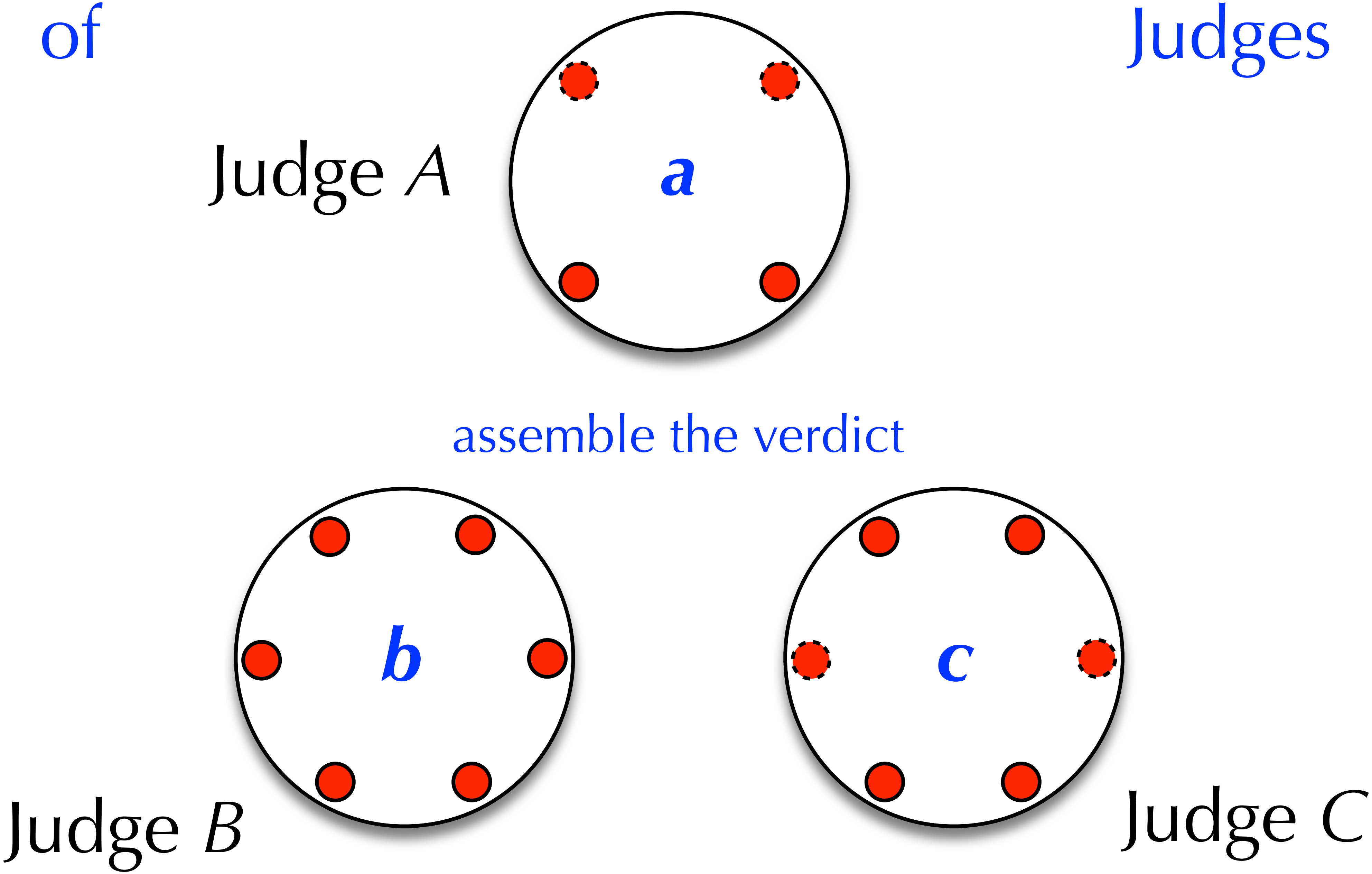
The tale of the Three Judges



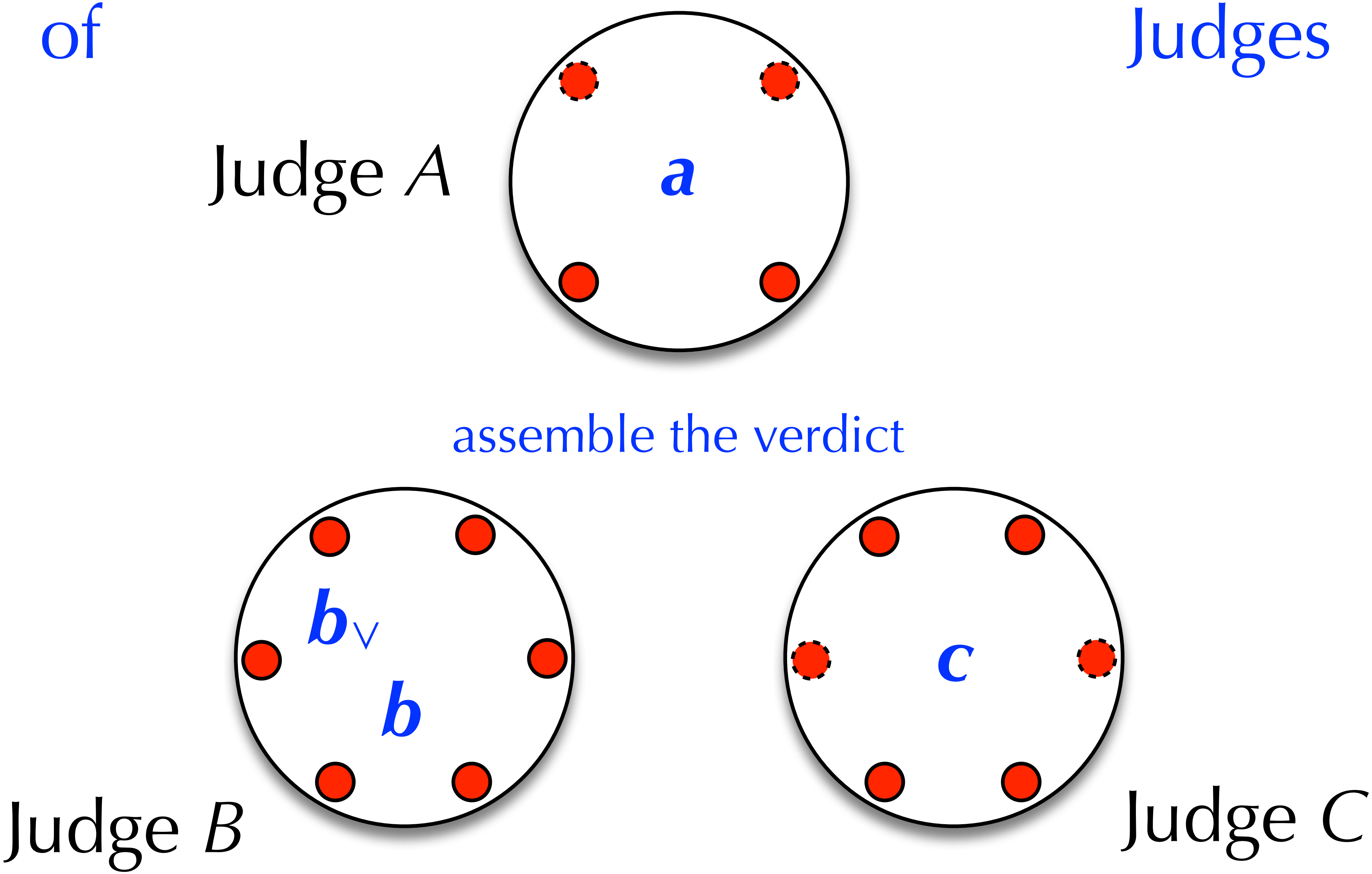
The tale of the Three Judges



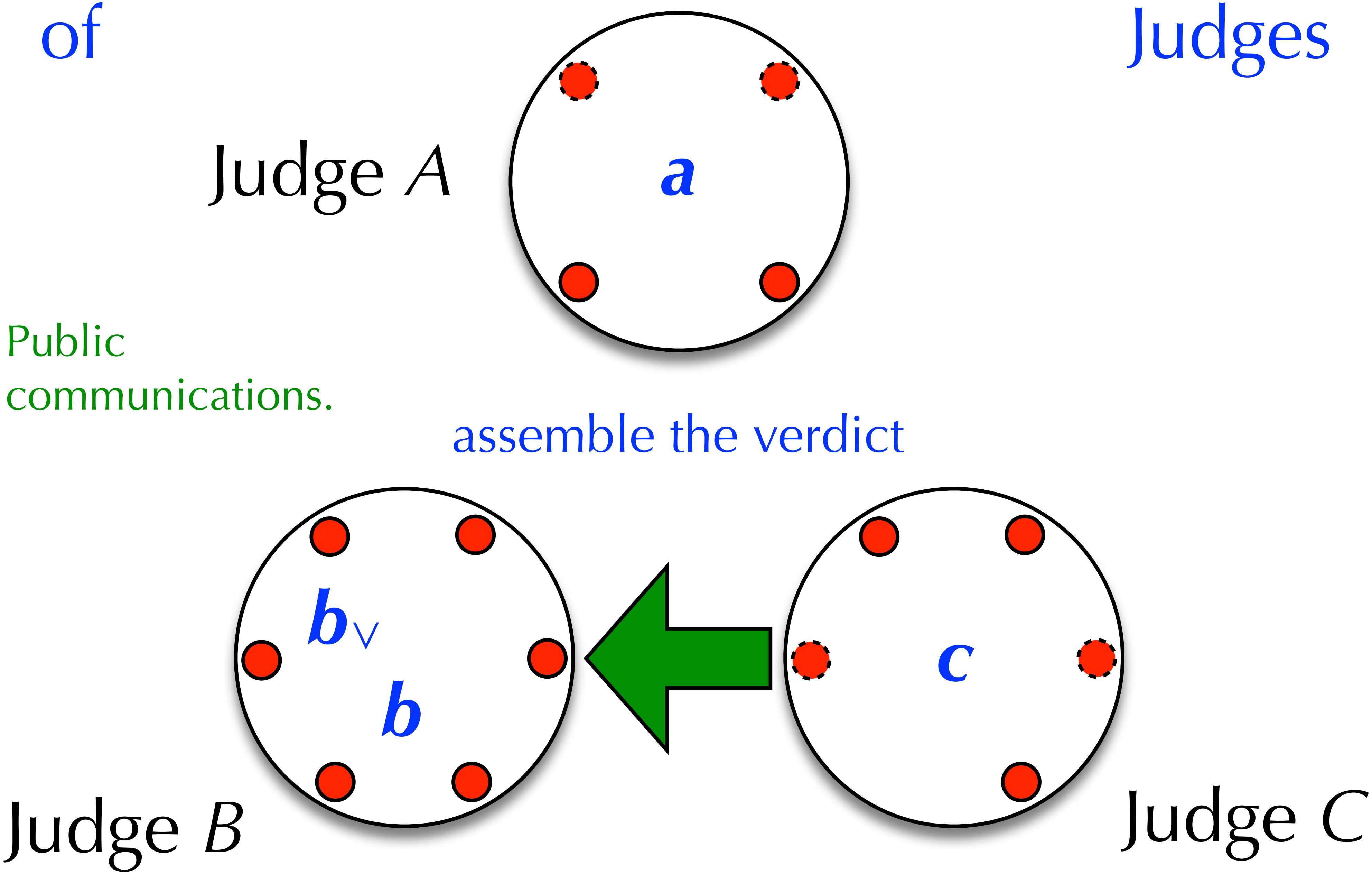
The tale of the Three Judges



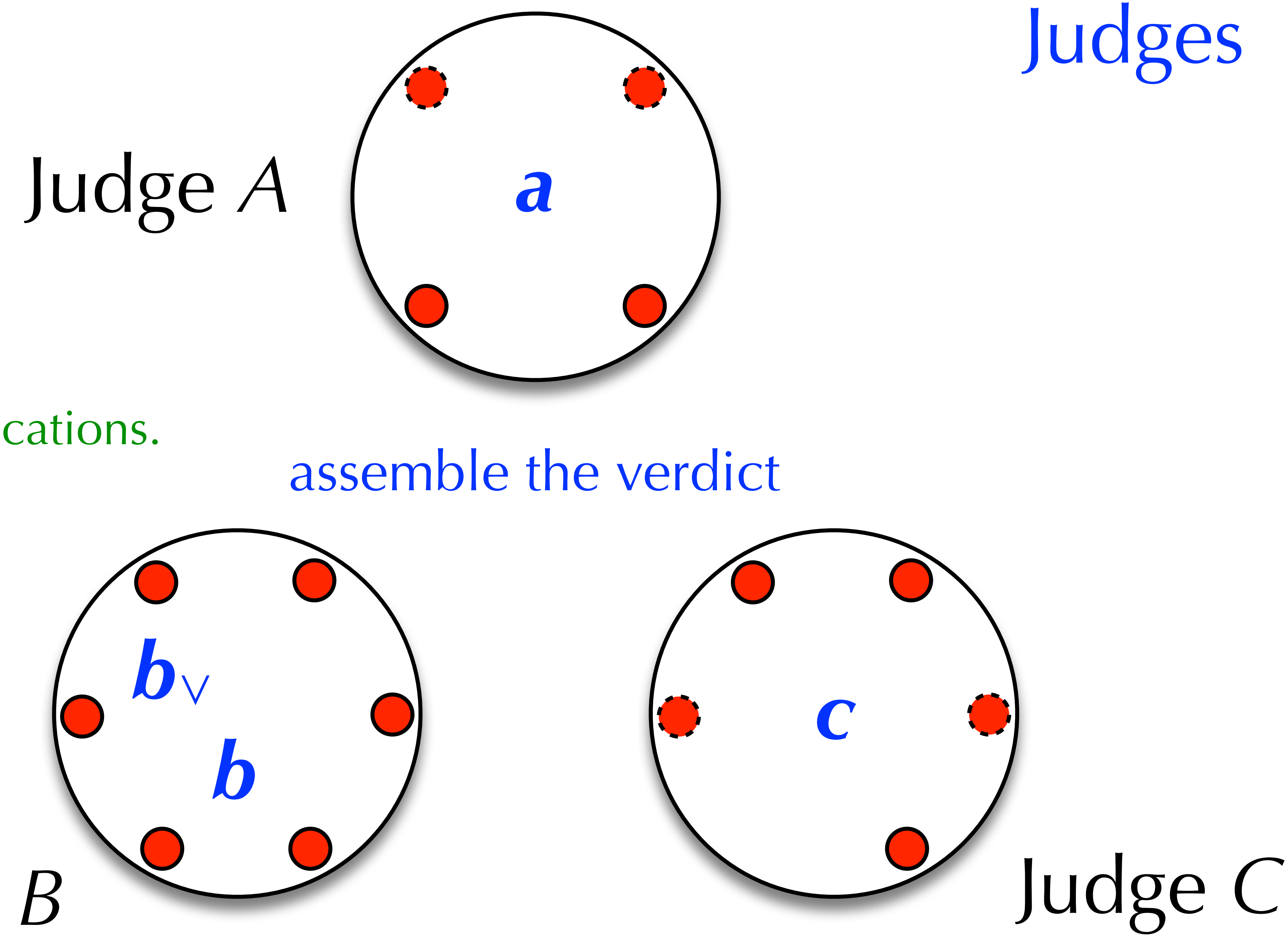
The tale of the Three Judges



The tale of the Three Judges



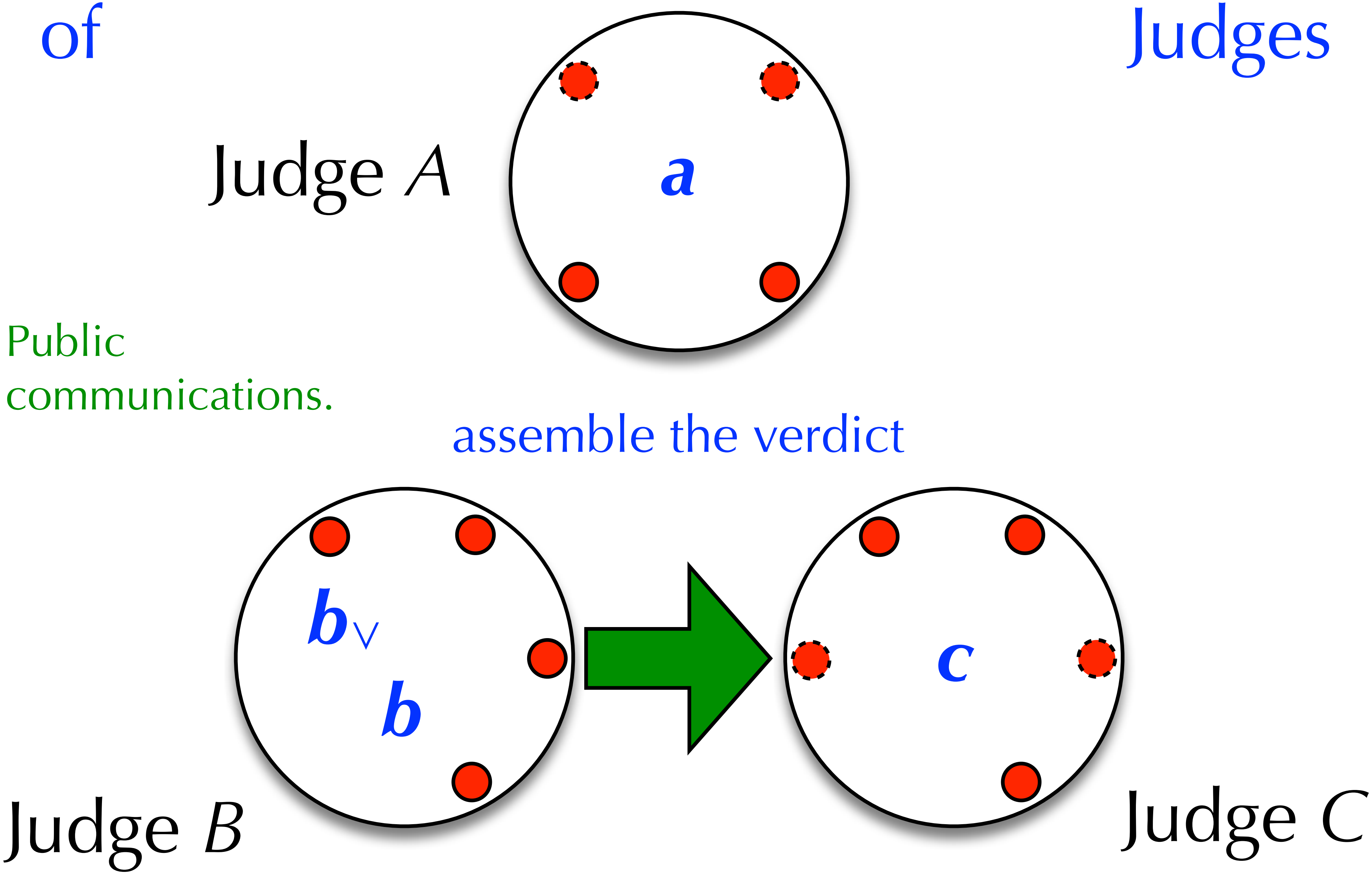
The tale of the Three Judges



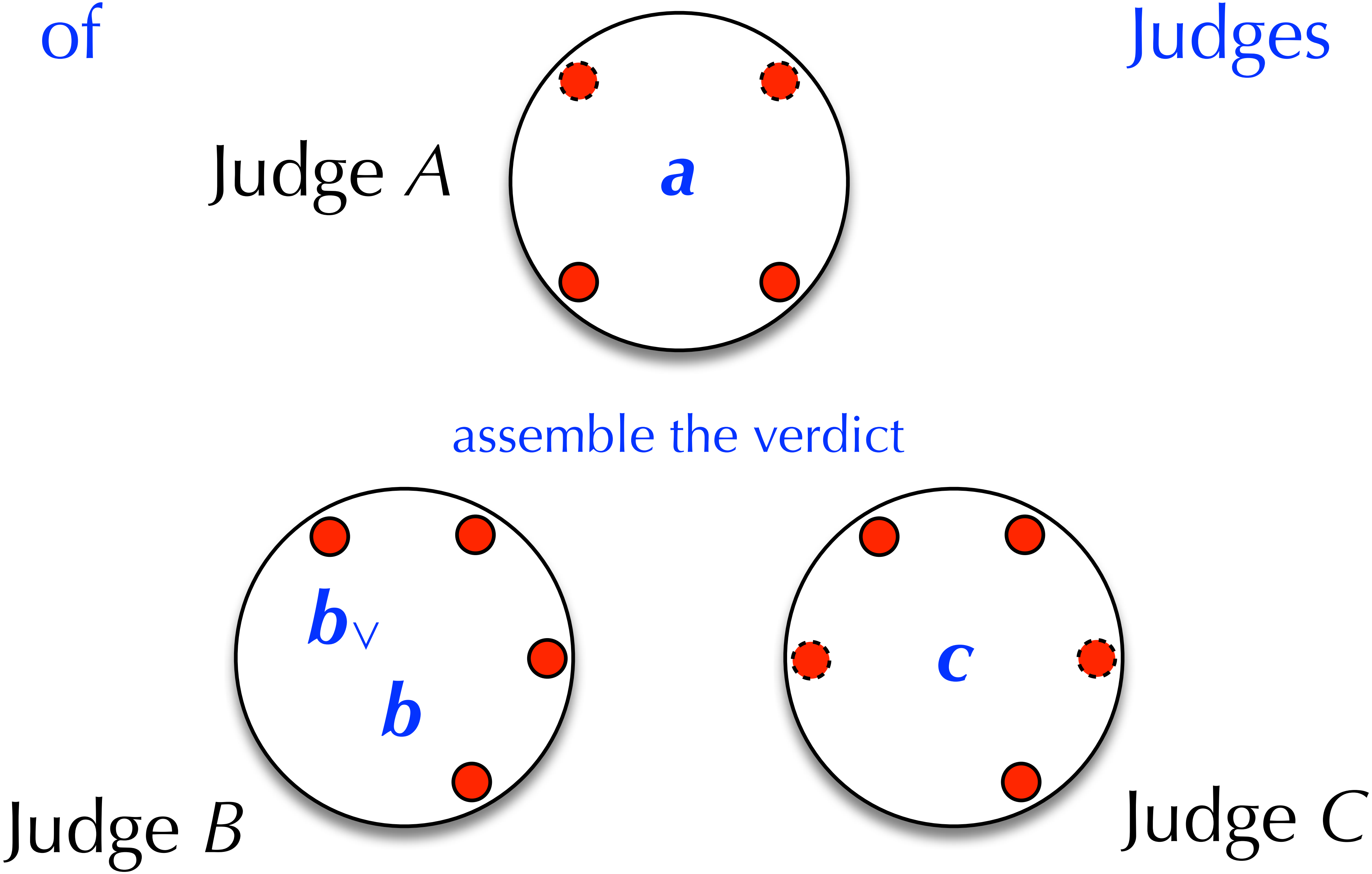
Public communications.

The tale of the Three Judges

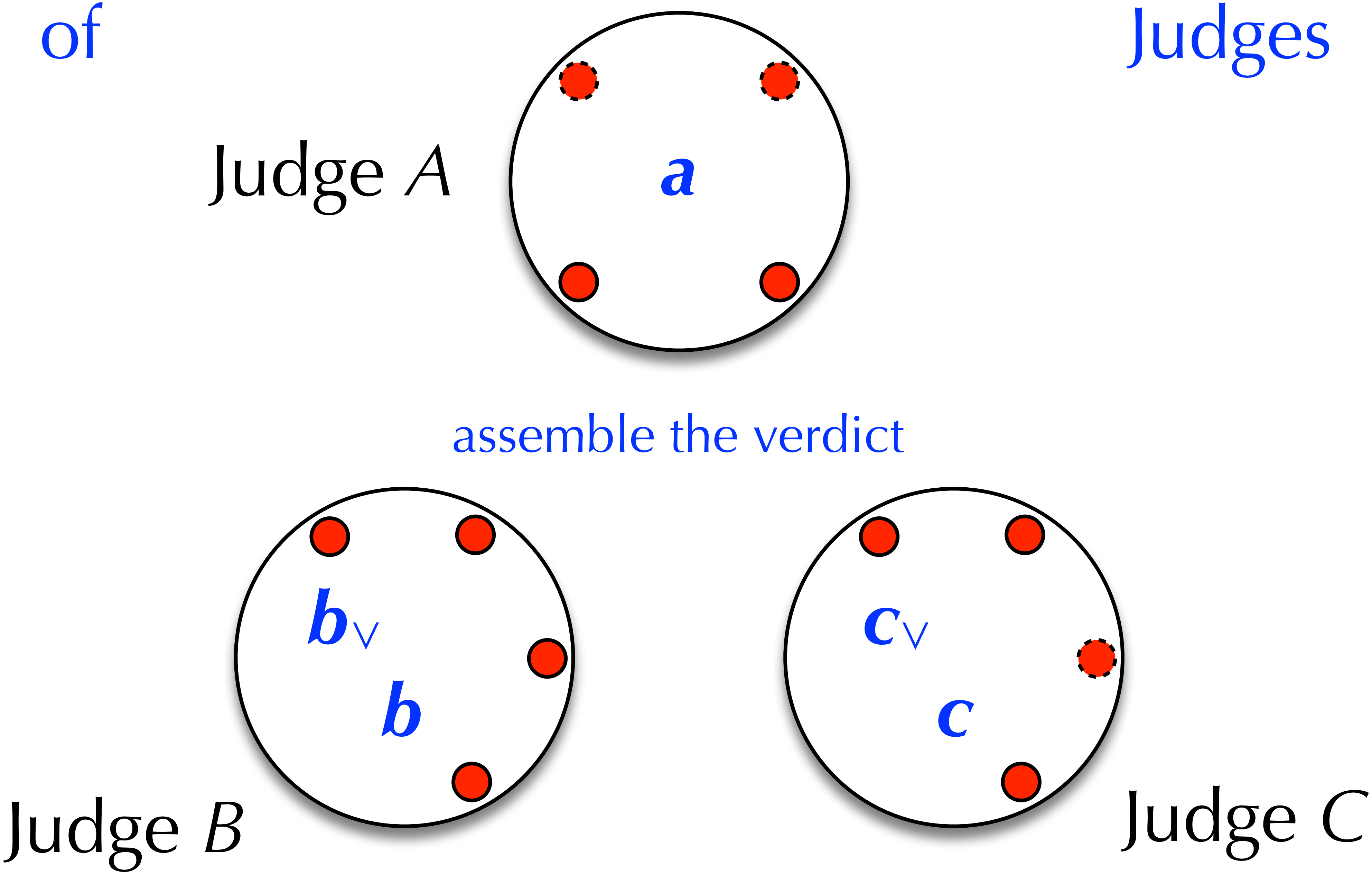
Public communications.



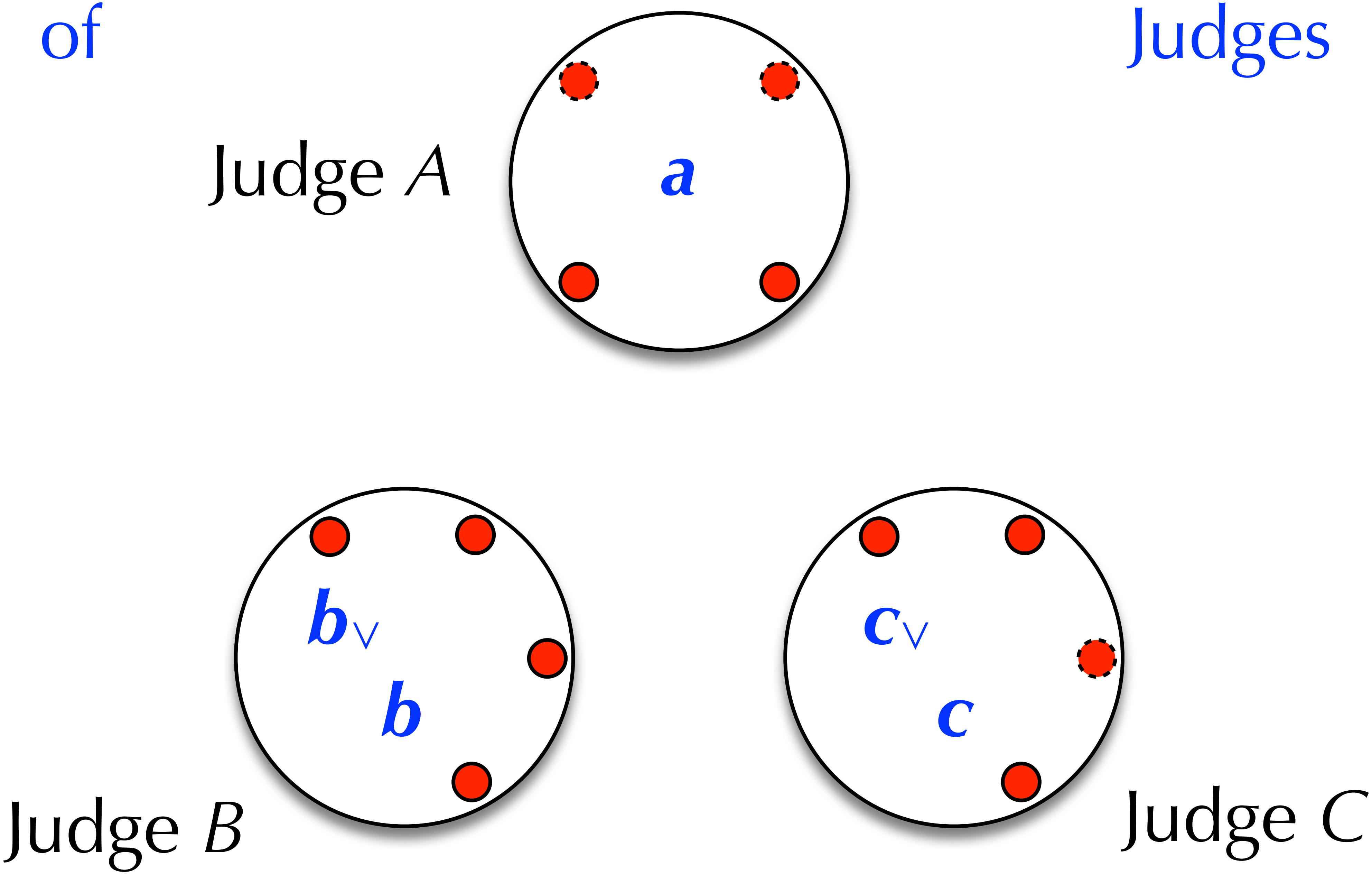
The tale of the Three Judges



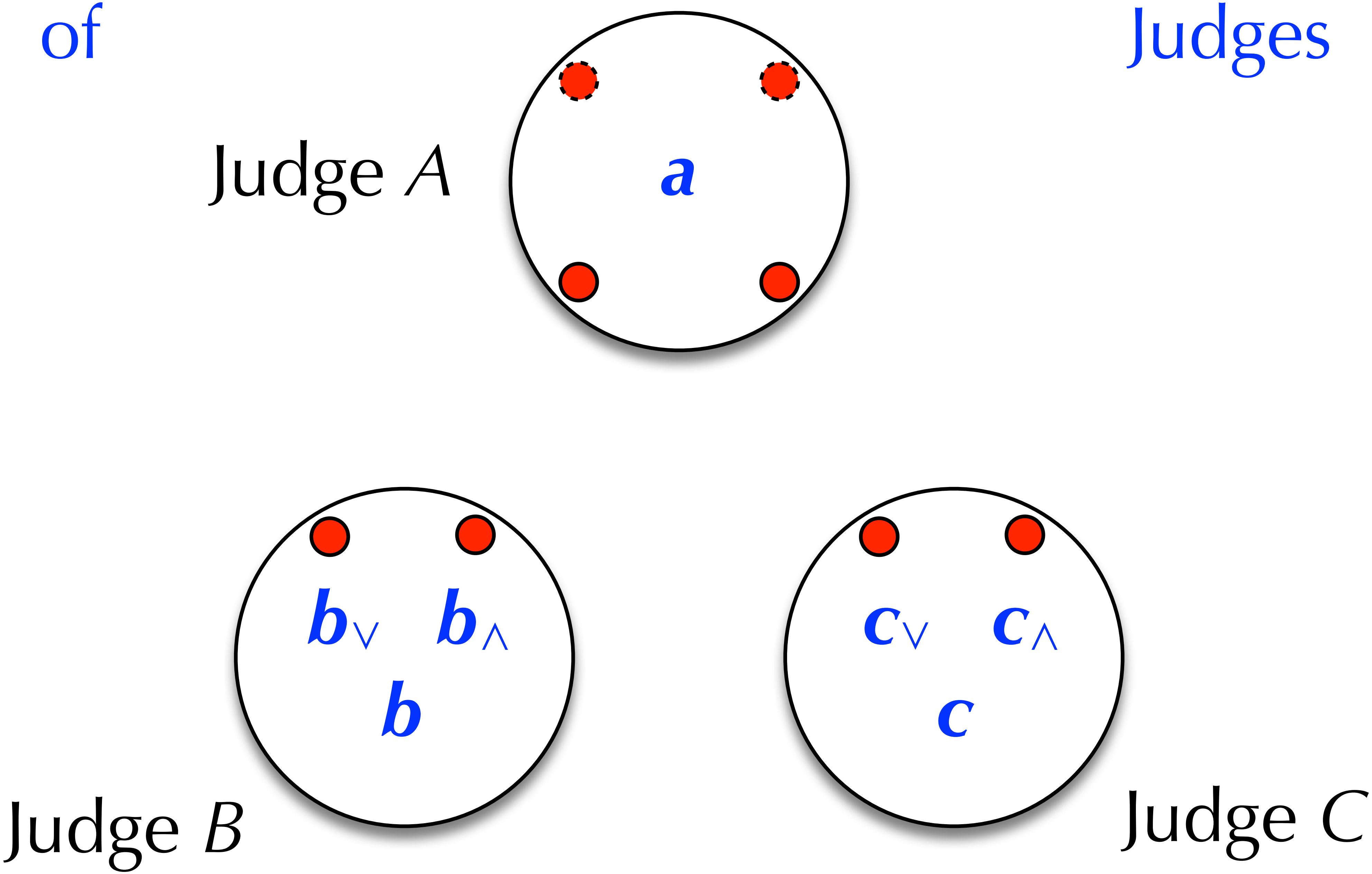
The tale of the Three Judges



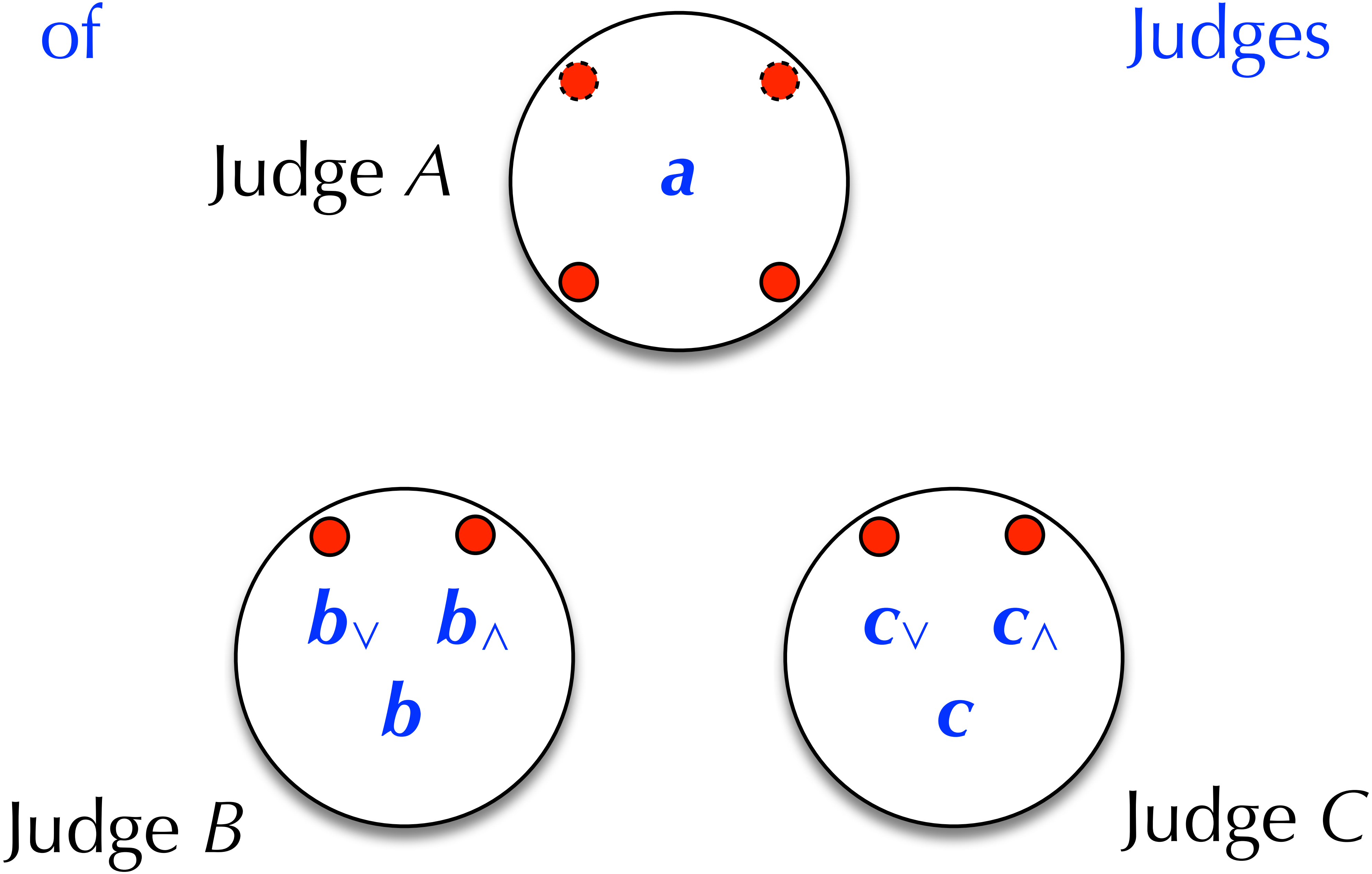
The tale of the Three Judges



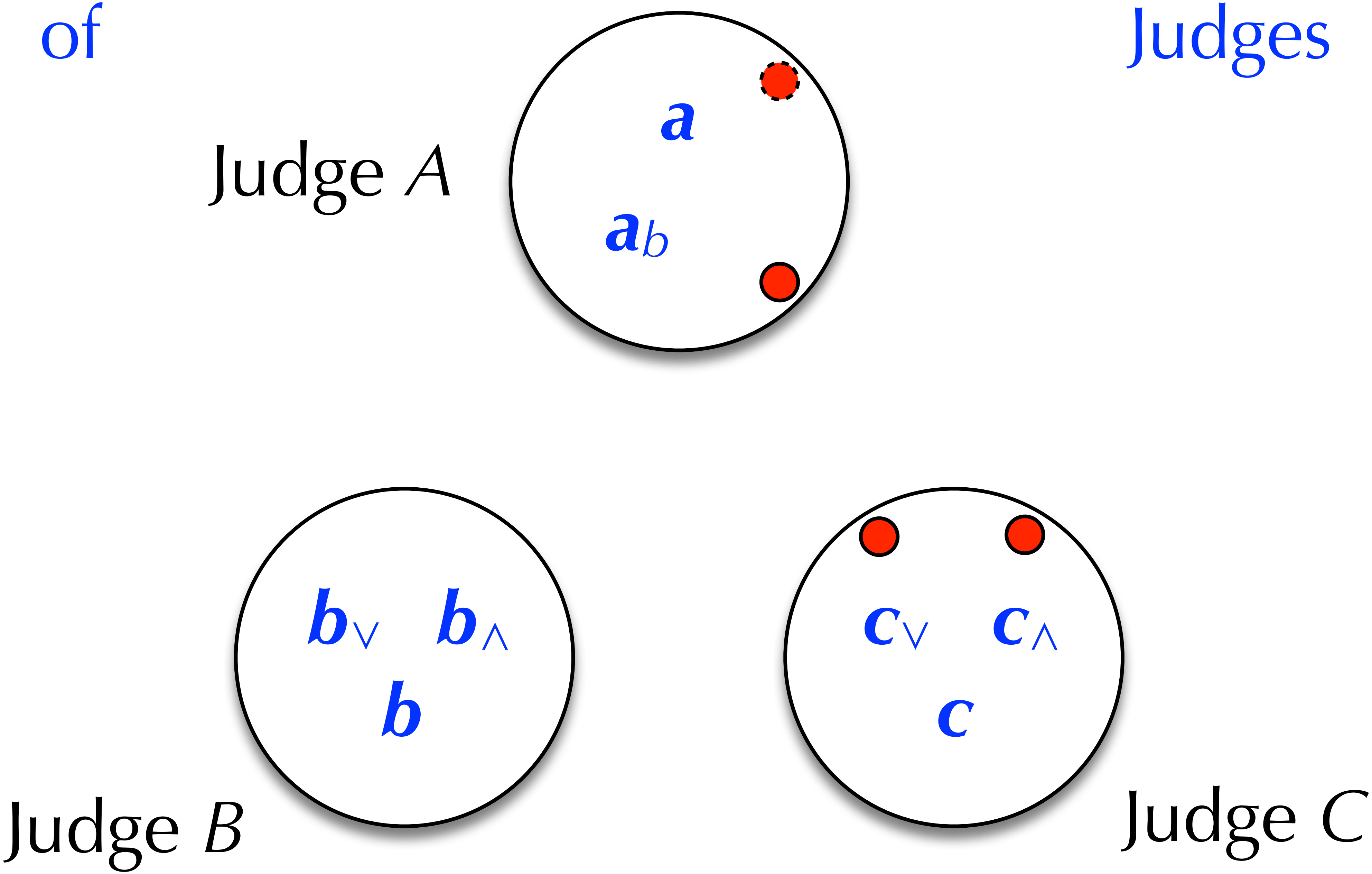
The tale of the Three Judges



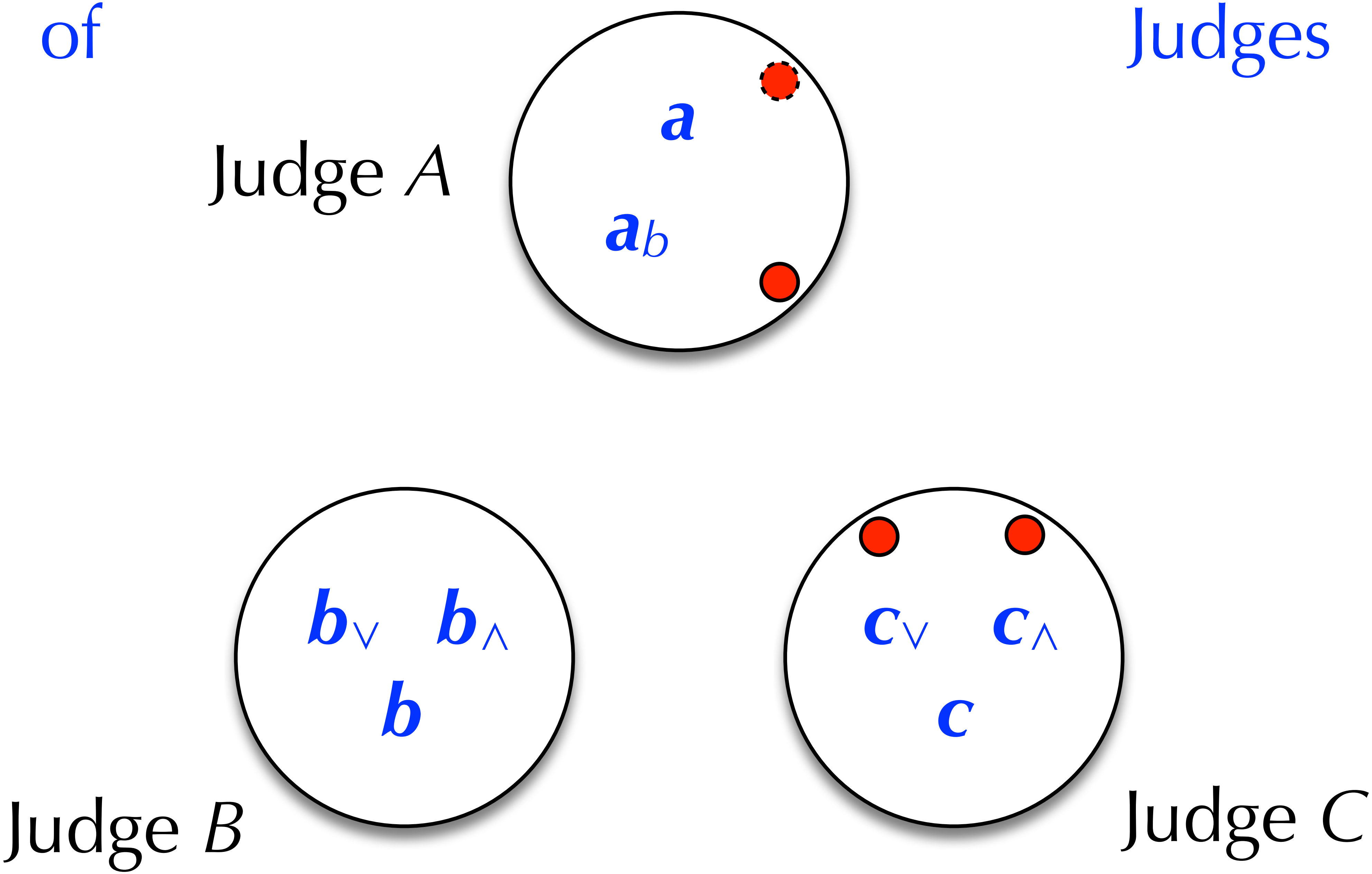
The tale of the Three Judges



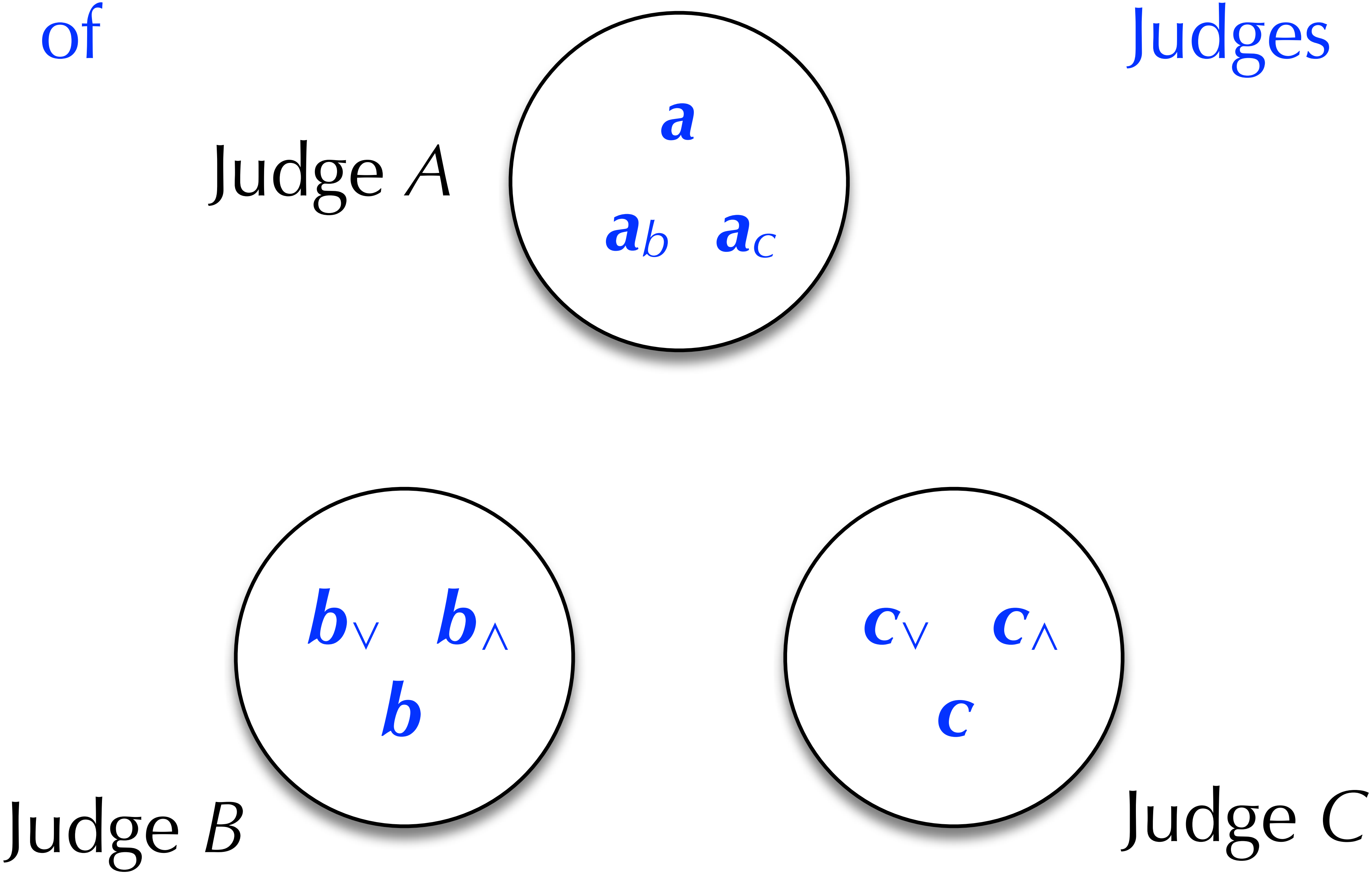
The tale of the Three Judges



The tale of the Three Judges



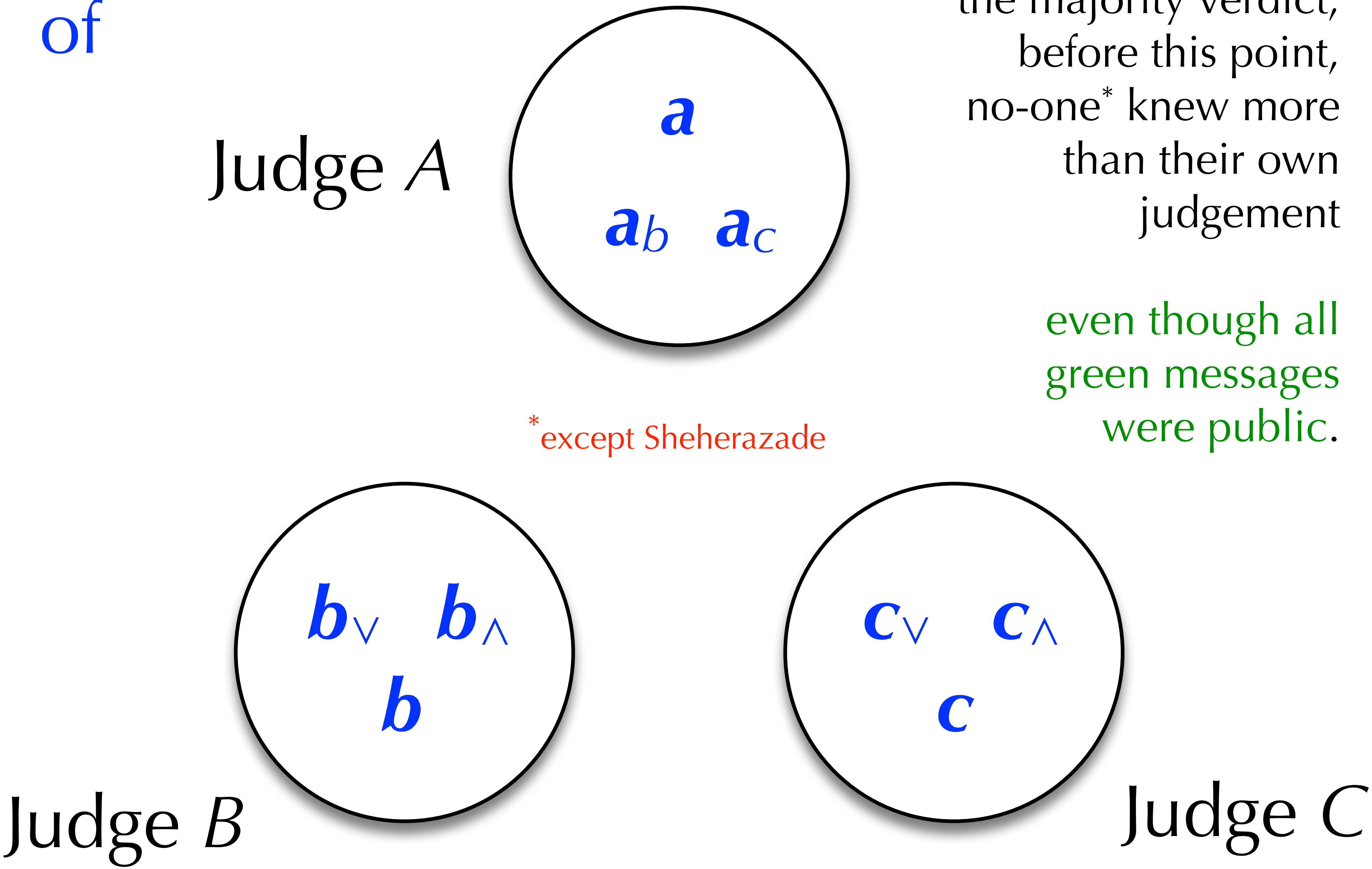
The tale of the Three Judges



The tale of

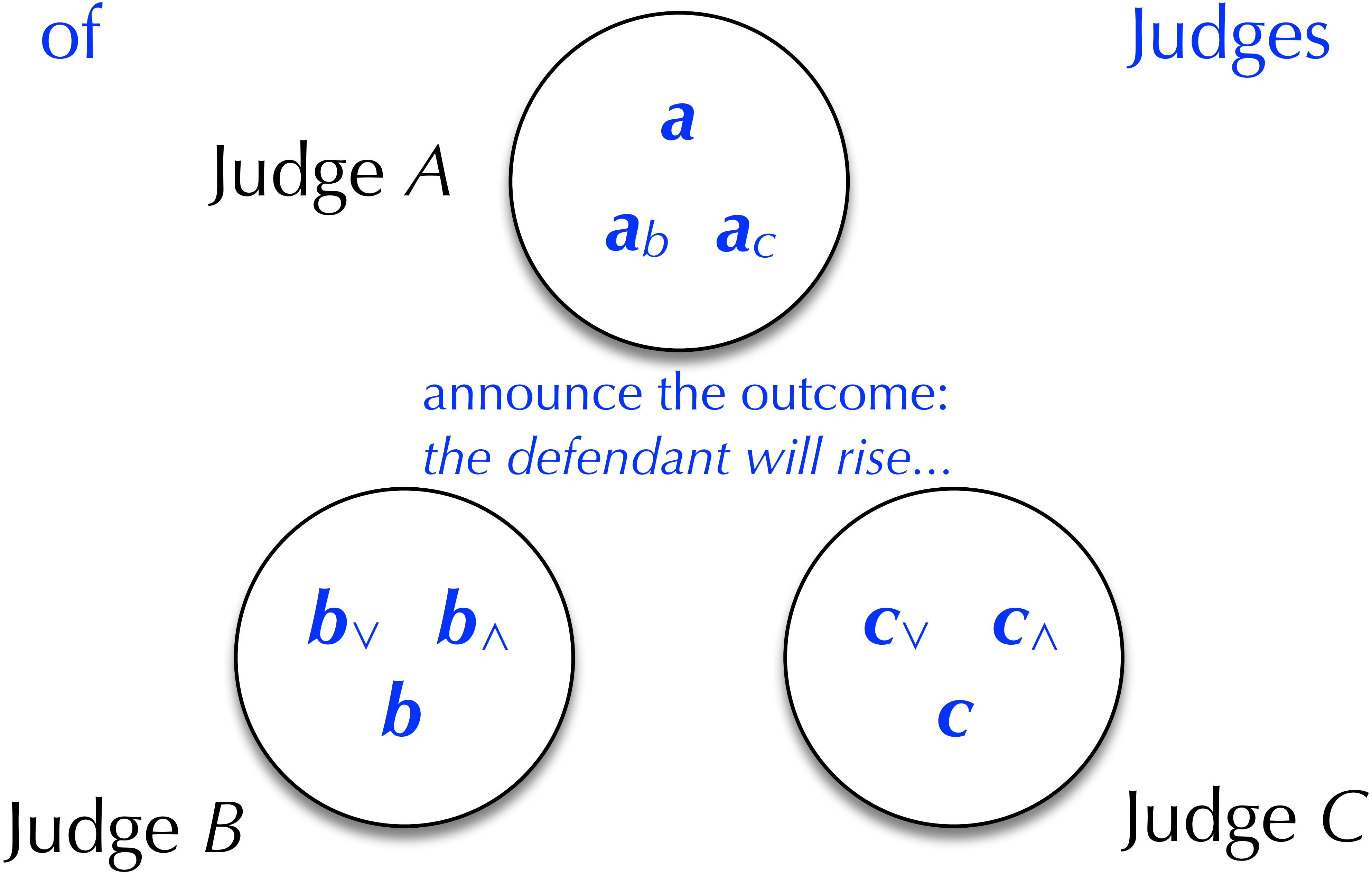
Judge A now knows the majority verdict; before this point, no-one* knew more than their own judgement

even though all green messages were public.

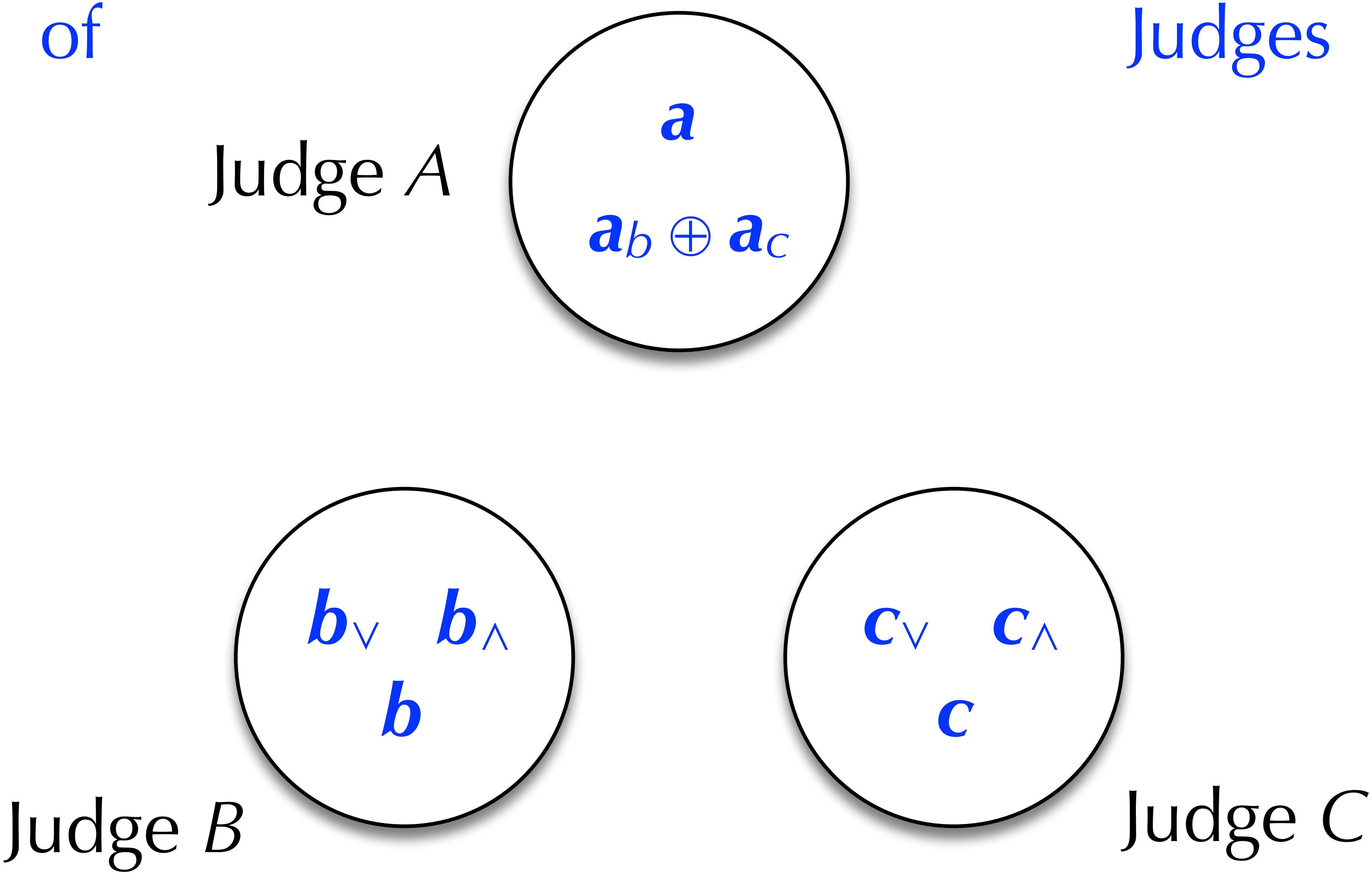


*except Sheherazade

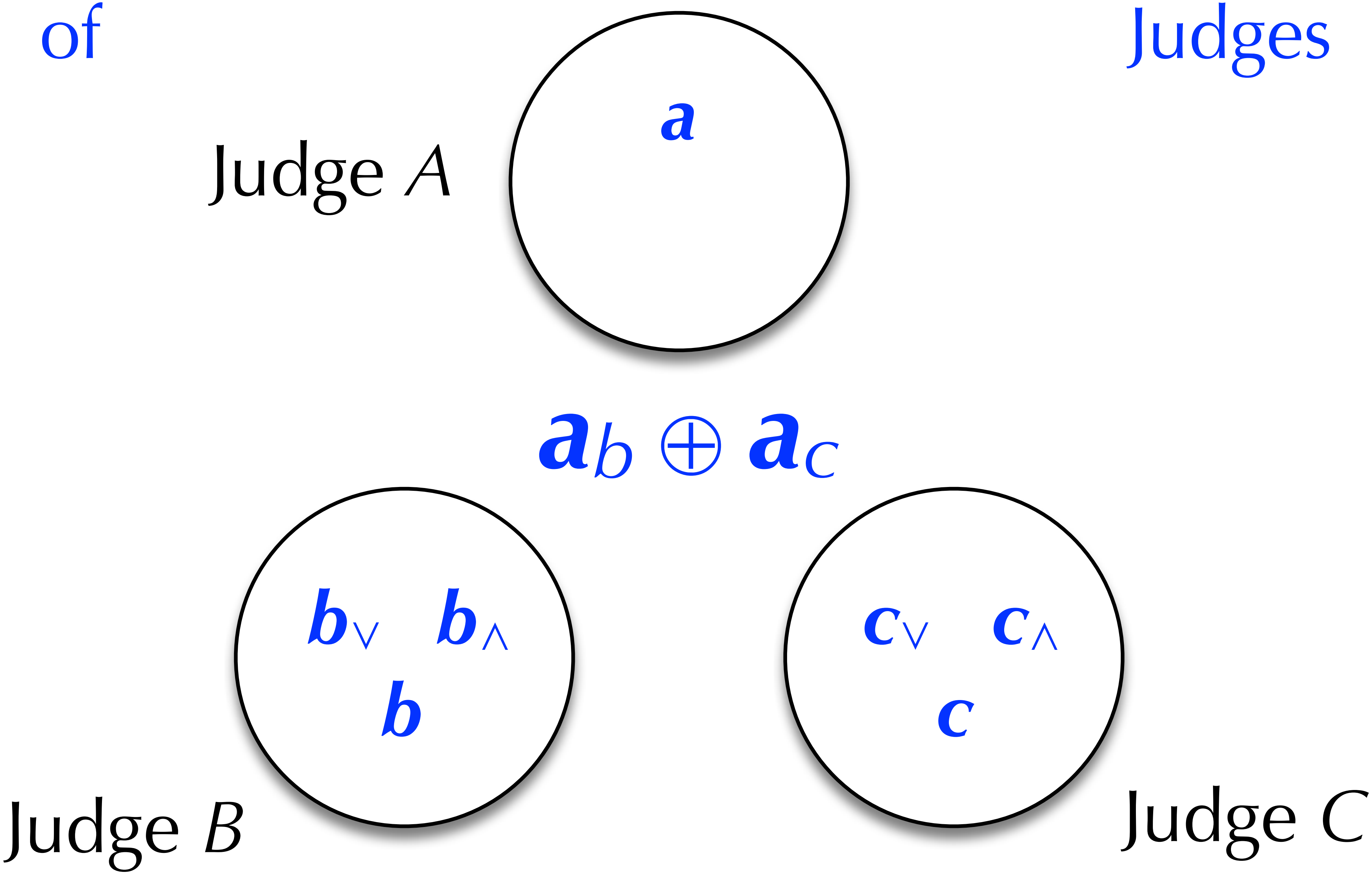
The tale of the Three Judges



The tale of the Three Judges



The tale of the Three Judges



The tale of the Three Judges

Judge A

$\boldsymbol{a}$

$\boldsymbol{a}_b$ $\boldsymbol{a}_c$

$$\boldsymbol{a}_b \oplus \boldsymbol{a}_c$$

$\boldsymbol{b}_\vee$ $\boldsymbol{b}_\wedge$

$\boldsymbol{b}$

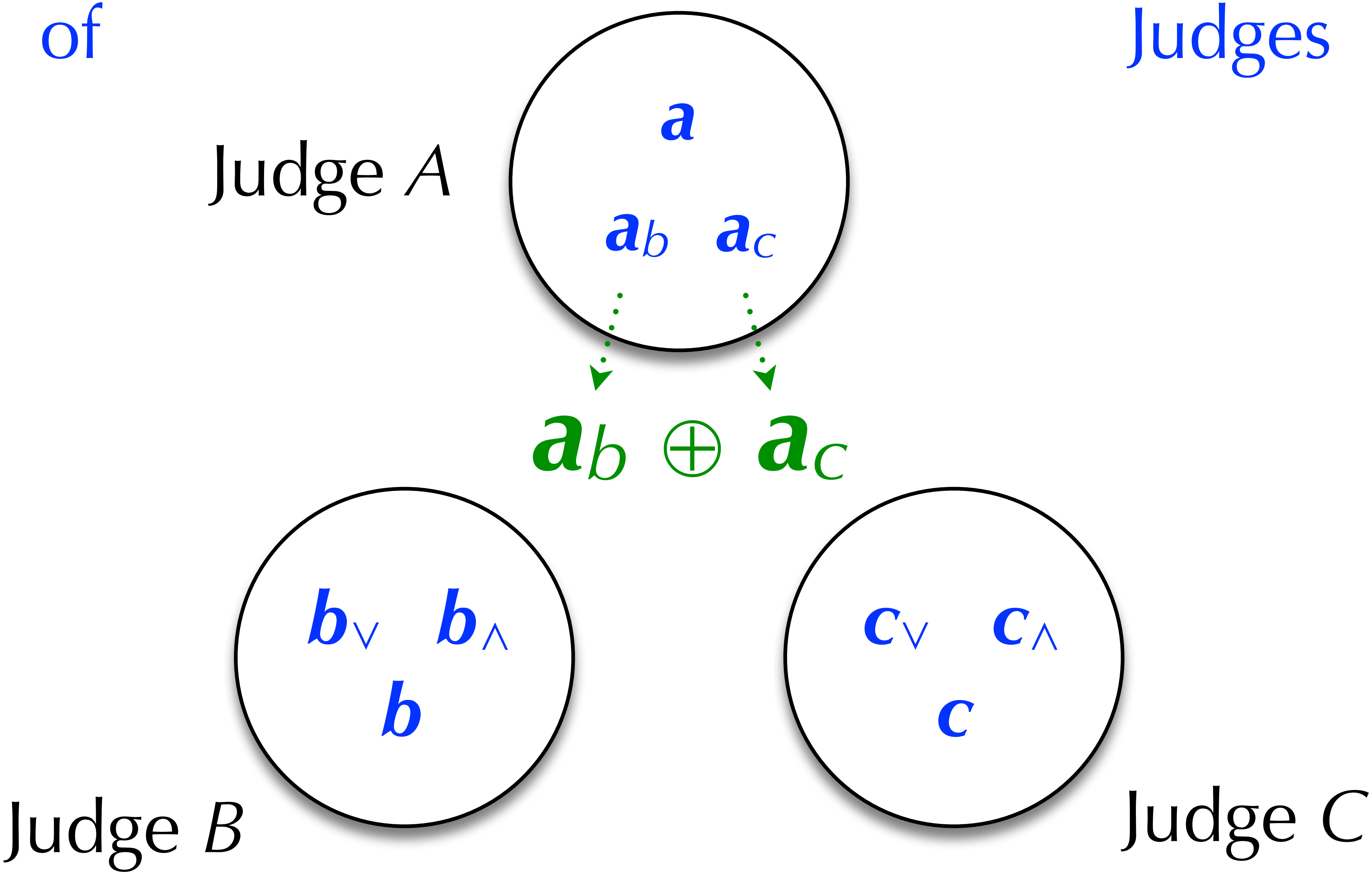
Judge B

$\boldsymbol{c}_\vee$ $\boldsymbol{c}_\wedge$

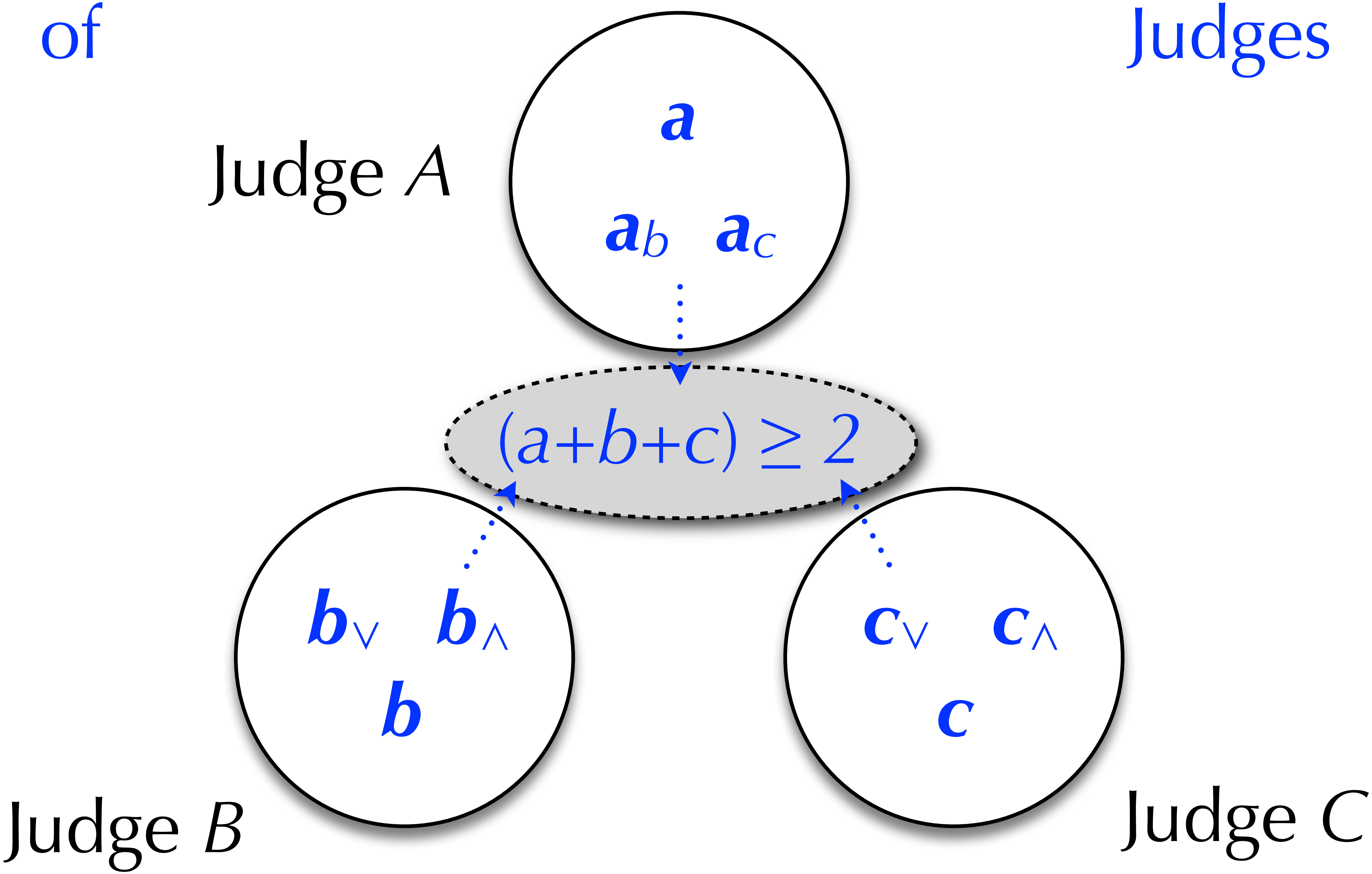
$\boldsymbol{c}$

Judge C

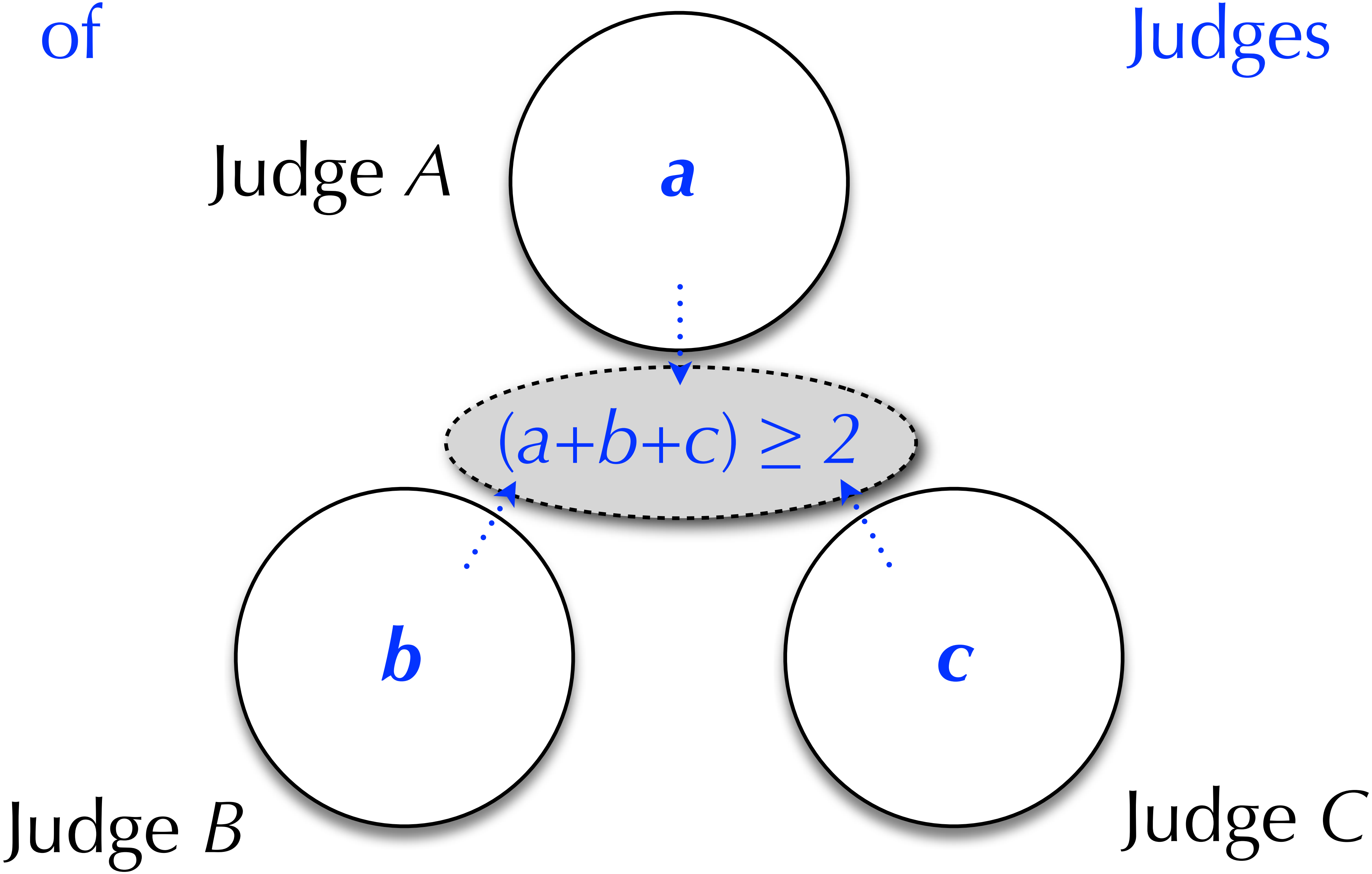
The tale of the Three Judges



The tale of the Three Judges



The tale of the Three Judges



Appendix

- Derivation of the Encryption Lemma
- Derivation of The Three Judges
- Derivation of Oblivious Transfer

The Encryption Lemma derivation

$$\textbf{var}\ \mathrm{Bool}\ b$$

$$\begin{array}{ll} |[& \textbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\ & b_1 \oplus b_2 := b \\ & \textbf{reveal}\ b_1 \\ ]| & \end{array}$$

The Encryption Lemma derivation

$$\mathbf{var}\ \mathrm{Bool}\ b$$

$$\begin{array}{ll} |[& \mathbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\ & b_1 \oplus b_2 := b \\ & \\ & \mathbf{reveal}\ b_1 \\ ]| & \end{array}$$

The Encryption Lemma derivation

$$
\begin{array}{l}
\textbf{var } \mathrm{Bool}\ b \\
\quad |[\quad \textbf{hid } \mathrm{Bool}\ b_1, b_2 \\
\qquad\quad b_2 :\in \{0,1\} \\
\qquad\quad b_1 := b \oplus b_2 \\
\qquad\quad \textbf{reveal } b_1 \\
\quad]|
\end{array}
$$

The Encryption Lemma derivation

$$\mathbf{var}\ \mathrm{Bool}\ b$$

$$
\begin{array}{ll}
|[& \mathbf{hid}\ \mathrm{Bool}\ b_1, b_2 \\
 & b_1 :\in \{0, 1\} \\
 & b_2 := b \oplus b_1 \\
 & \mathbf{reveal}\ b_1 \\
]| &
\end{array}
$$

The Encryption Lemma derivation

$$\textbf{var}\ \text{Bool}\ b$$

$$
\begin{array}{ll}
|[& \textbf{hid}\ \text{Bool}\ b_1, b_2 \\
 & b_1 :\in \{0,1\} \\
 & b_2 := b \oplus b_1 \\
 & \textbf{reveal}\ b_1 \\
]| &
\end{array}
$$

The Encryption Lemma derivation

$$
\begin{array}{l}
\textbf{var}\ \mathrm{Bool}\ b \\
\quad |[\quad \textbf{hid}\ \mathrm{Bool}\ b_1 \\
\qquad\quad b_1 :\in \{0,1\} \\
\qquad\quad \textbf{reveal}\ b_1 \\
\quad]|
\end{array}
$$

The Encryption Lemma derivation

$$\mathbf{var}\ \mathrm{Bool}\ b$$

$$\quad \mathbf{skip}$$

Appendix

- Derivation of the Encryption Lemma
- Derivation of The Three Judges
- Derivation of Oblivious Transfer

The Three-Judges derivation

$$\begin{array}{l} \mathbf{vis}_A\ a \\ \mathbf{vis}_B\ b \\ \mathbf{vis}_C\ c \\ \quad \mathbf{reveal}\ (a{+}b{+}c \geq 2) \end{array}$$

The Three-Judges derivation

$$\begin{array}{l} \mathbf{vis}_A\ a \\ \mathbf{vis}_B\ b \\ \mathbf{vis}_C\ c \end{array}$$

$$\mathbf{reveal}\ (a{+}b{+}c \geq 2)$$

The Three-Judges derivation

$$\mathbf{vis}_A\ a$$
$$\mathbf{vis}_B\ b$$
$$\mathbf{vis}_C\ c$$

$$|[\quad \mathbf{vis}_B\ b_\wedge ; \mathbf{vis}_C\ c_\wedge$$

Lovers' Protocol

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$]|$$

$$\mathbf{reveal}\ (a{+}b{+}c \geq 2)$$

A: trivial.
B,*C*: EL.

The Three-Judges derivation

$$
\begin{array}{l}
\mathbf{vis}_A\ a \\
\mathbf{vis}_B\ b \\
\mathbf{vis}_C\ c \\
|[\ \mathbf{vis}_B\ b_\wedge ; \mathbf{vis}_C\ c_\wedge
\end{array}
$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$\mathbf{reveal}\ (a{+}b{+}c \geq 2)$$

The Three-Judges derivation

$$\mathbf{vis}_A\ a$$
$$\mathbf{vis}_B\ b$$
$$\mathbf{vis}_C\ c$$
$$[\![\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge$$
$$\mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$b_\vee \oplus c_\vee := b \vee c$$

$$\mathbf{reveal}\ (a+b+c \geq 2)$$

A: trivial.

B,*C*: EL.

The Three-Judges derivation

$$
\begin{array}{l}
\mathbf{vis}_A \ a \\
\mathbf{vis}_B \ b \\
\mathbf{vis}_C \ c \\
|[\ \mathbf{vis}_B \ b_\wedge ; \mathbf{vis}_C \ c_\wedge \\
\quad \mathbf{vis}_B \ b_\vee ; \mathbf{vis}_C \ c_\vee \\
\quad \mathbf{vis}_A \ a_b, a_c
\end{array}
$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b \oplus a_c := (a{+}b{+}c \geq 2)$$
$$\mathbf{reveal} \ a_b \oplus a_c$$

A: reveal.
B,*C*: trivial.

The Three-Judges derivation

$$
\begin{array}{l}
\mathbf{vis}_A\ a \\
\mathbf{vis}_B\ b \\
\mathbf{vis}_C\ c \\
|[\ \mathbf{vis}_B\ b_\wedge ; \mathbf{vis}_C\ c_\wedge \\
\quad \mathbf{vis}_B\ b_\vee ; \mathbf{vis}_C\ c_\vee \\
\quad \mathbf{vis}_A\ a_b, a_c
\end{array}
$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b \oplus a_c := (b_\vee \oplus c_\vee \ \lhd a \rhd \ b_\wedge \oplus c_\wedge)$$

$$\mathbf{reveal}\ a_b \oplus a_c$$

prop calc

prog alg

The Three-Judges derivation

$$
\begin{array}{l}
\mathbf{vis}_A\ a \\
\mathbf{vis}_B\ b \\
\mathbf{vis}_C\ c \\
|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge \\
\quad \mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee \\
\quad \mathbf{vis}_A\ a_b, a_c
\end{array}
$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b \oplus a_c := (b_\vee \lhd a \rhd b_\wedge) \oplus (c_\vee \lhd a \rhd c_\wedge)$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

prop calc

The Three-Judges derivation

$$
\begin{array}{l}
\mathbf{vis}_A\ a \\
\mathbf{vis}_B\ b \\
\mathbf{vis}_C\ c \\
\|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge \\
\quad \mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee \\
\quad \mathbf{vis}_A\ a_b, a_c
\end{array}
$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

The Three-Judges derivation

This resolution of nondeterminism is *not valid* on its own.

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$
\begin{array}{l}
\mathbf{vis}_A\ a \\
\mathbf{vis}_B\ b \\
\mathbf{vis}_C\ c \\
|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge \\
\quad \mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee \\
\quad \mathbf{vis}_A\ a_b, a_c
\end{array}
$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

The Three-Judges derivation

This resolution of nondeterminism is *not valid* on its own.

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b \oplus a_c := (b_\vee \lhd a \rhd b_\wedge) \oplus (c_\vee \lhd a \rhd c_\wedge)$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$

$$\textbf{reveal } a_b \oplus a_c$$

$\textbf{vis}_A\ a$
$\textbf{vis}_B\ b$
$\textbf{vis}_C\ c$
$\|[\ \textbf{vis}_B\ b_\wedge; \textbf{vis}_C\ c_\wedge$
$\quad \textbf{vis}_B\ b_\vee; \textbf{vis}_C\ c_\vee$
$\quad \textbf{vis}_A\ a_b, a_c$

The Three-Judges derivation

This resolution of nondeterminism is *not valid* on its own.

For *B*,*C* it is invalid because it resolves hidden nondeterminism in the *A*-variables.

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b \oplus a_c := (b_\vee \lhd a \rhd b_\wedge) \oplus (c_\vee \lhd a \rhd c_\wedge)$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$

$$\textbf{reveal } a_b \oplus a_c$$

The Three-Judges derivation

This resolution of nondeterminism is *not valid* on its own.

For *B*,*C* it is invalid because it resolves hidden nondeterminism in the *A*-variables.

For *A* it is invalid because it reveals information about the separate halves of the exclusive-or.

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$\textbf{reveal } a_b \oplus a_c$$

The Three-Judges derivation

$\boxed{\textit{B,C's point of view.}}$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$\mathbf{vis}_A\ a$
$\mathbf{vis}_B\ b$
$\mathbf{vis}_C\ c$
$|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge$
$\quad \mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee$
$\quad \mathbf{vis}_A\ a_b, a_c$

$$a_b \oplus a_c := (b_\vee \lhd a \rhd b_\wedge) \oplus (c_\vee \lhd a \rhd c_\wedge)$$

$$\mathbf{reveal}\ a_b \oplus a_c$$

The Three-Judges derivation

B,C's point of view.

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$b_\vee \oplus c_\vee := b \vee c$$

$$
\begin{array}{l}
\textbf{vis}_A\ a \\
\textbf{vis}_B\ b \\
\textbf{vis}_C\ c \\
|[\ \textbf{vis}_B\ b_\wedge; \textbf{vis}_C\ c_\wedge \\
\quad \textbf{vis}_B\ b_\vee; \textbf{vis}_C\ c_\vee \\
\quad \textbf{vis}_A\ a_b, a_c
\end{array}
$$

$$a_b \oplus a_c := (b_\vee \lhd a \rhd b_\wedge) \oplus (c_\vee \lhd a \rhd c_\wedge)$$

$$a_b \oplus a_c := a_b \oplus a_c$$

$$\textbf{reveal}\ a_b \oplus a_c$$

The Three-Judges derivation

$\boxed{\textit{B,C's point of view.}}$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$\begin{array}{l}
\mathbf{vis}_A\ a \\
\mathbf{vis}_B\ b \\
\mathbf{vis}_C\ c \\
|[\ \mathbf{vis}_B\ b_\wedge ; \mathbf{vis}_C\ c_\wedge \\
\quad \mathbf{vis}_B\ b_\vee ; \mathbf{vis}_C\ c_\vee \\
\quad \mathbf{vis}_A\ a_b, a_c
\end{array}$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$a_b \oplus a_c := a_b \oplus a_c$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

The Three-Judges derivation

B,C's point of view.

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$
\begin{array}{l}
\textbf{vis}_A\ a \\
\textbf{vis}_B\ b \\
\textbf{vis}_C\ c \\
|[\ \textbf{vis}_B\ b_\wedge; \textbf{vis}_C\ c_\wedge \\
\quad \textbf{vis}_B\ b_\vee; \textbf{vis}_C\ c_\vee \\
\quad \textbf{vis}_A\ a_b, a_c
\end{array}
$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$\textbf{reveal}\ a_b \oplus a_c$$
$$a_b \oplus a_c := a_b \oplus a_c\ ]|$$

The Three-Judges derivation

B,C's point of view.

$$
\begin{array}{l}
\mathbf{vis}_A\ a \\
\mathbf{vis}_B\ b \\
\mathbf{vis}_C\ c \\
|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge \\
\quad \mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee \\
\quad \mathbf{vis}_A\ a_b, a_c
\end{array}
$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

The Three-Judges derivation

A's point of view,
when a is true.

$\mathbf{vis}_A\ a$
$\mathbf{vis}_B\ b$
$\mathbf{vis}_C\ c$
$|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge$
$\quad \mathbf{vis}_{AB}\ b_\vee; \mathbf{vis}_{AC}\ c_\vee$
$\quad \mathbf{vis}_A\ a_b, a_c$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b \oplus a_c := (b_\vee \lhd a \rhd b_\wedge) \oplus (c_\vee \lhd a \rhd c_\wedge)$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

A: reveal
not "trivial"

The Three-Judges derivation

A's point of view,
when *a* is true.

$$\mathbf{vis}_A\ a$$
$$\mathbf{vis}_B\ b$$
$$\mathbf{vis}_C\ c$$
$$|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge$$
$$\mathbf{vis}_{AB}\ b_\vee; \mathbf{vis}_{AC}\ c_\vee$$
$$\mathbf{vis}_A\ a_b, a_c$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

A: standard
refinement

The Three-Judges derivation

A's point of view, when *a* is true.

$$\mathbf{vis}_A\ a$$
$$\mathbf{vis}_B\ b$$
$$\mathbf{vis}_C\ c$$
$$|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge$$
$$\mathbf{vis}_{AB}\ b_\vee; \mathbf{vis}_{AC}\ c_\vee$$
$$\mathbf{vis}_A\ a_b, a_c$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b \oplus a_c := (b_\vee \lhd a \rhd b_\wedge) \oplus (c_\vee \lhd a \rhd c_\wedge)$$
$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

A: standard refinement

The Three-Judges derivation

A's point of view,
when *a* is true.

$$\mathbf{vis}_A\ a$$
$$\mathbf{vis}_B\ b$$
$$\mathbf{vis}_C\ c$$
$$|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge$$
$$\mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee$$
$$\mathbf{vis}_A\ a_b, a_c$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

A: reduce
visibility

The Three-Judges derivation

A's point of view,
when *a* is true and,
by symmetry,
false too.

$\mathbf{vis}_A\ a$
$\mathbf{vis}_B\ b$
$\mathbf{vis}_C\ c$
$|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge$
$\quad \mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee$
$\quad \mathbf{vis}_A\ a_b, a_c$

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b, a_c := (b_\vee \triangleleft a \triangleright b_\wedge), (c_\vee \triangleleft a \triangleright c_\wedge)$$

$$\mathbf{reveal}\ a_b \oplus a_c$$

A: symmetry

The Three-Judges derivation

An alternative is to consider the three statements together.

$$\mathbf{vis}_A\ a$$
$$\mathbf{vis}_B\ b$$
$$\mathbf{vis}_C\ c$$
$$|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge$$
$$\mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee$$
$$\mathbf{vis}_A\ a_b, a_c$$

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b \oplus a_c := (b_\vee \triangleleft a \triangleright b_\wedge) \oplus (c_\vee \triangleleft a \triangleright c_\wedge)$$
$$\mathbf{reveal}\ a_b \oplus a_c$$

The Three-Judges derivation

The case-analysis is avoided; but it becomes less algebraic.

$$b_\wedge \oplus c_\wedge := b \wedge c$$

$$b_\vee \oplus c_\vee := b \vee c$$

$$\begin{array}{l}
\mathbf{vis}_A\ a \\
\mathbf{vis}_B\ b \\
\mathbf{vis}_C\ c \\
|[\ \mathbf{vis}_B\ b_\wedge; \mathbf{vis}_C\ c_\wedge \\
\quad \mathbf{vis}_B\ b_\vee; \mathbf{vis}_C\ c_\vee \\
\quad \mathbf{vis}_A\ a_b, a_c
\end{array}$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$

$$\mathbf{reveal}\ a_b \oplus a_c$$

macro-atomicity

The Three-Judges derivation

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$\begin{array}{l}
\textbf{vis}_A\ a \\
\textbf{vis}_B\ b \\
\textbf{vis}_C\ c \\
|[\ \textbf{vis}_B\ b_\wedge; \textbf{vis}_C\ c_\wedge \\
\quad \textbf{vis}_B\ b_\vee; \textbf{vis}_C\ c_\vee \\
\quad \textbf{vis}_A\ a_b, a_c
\end{array}$$

$$a_b, a_c := (b_\vee \lhd a \rhd b_\wedge), (c_\vee \lhd a \rhd c_\wedge)$$
$$\textbf{reveal}\ a_b \oplus a_c$$

The Three-Judges derivation

$$b_\wedge \oplus c_\wedge := b \wedge c$$
$$b_\vee \oplus c_\vee := b \vee c$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$
$$a_c := (c_\vee \lhd a \rhd c_\wedge)$$

$$\textbf{reveal}\ a_b \oplus a_c$$

$$\begin{array}{l}
\textbf{vis}_A\ a \\
\textbf{vis}_B\ b \\
\textbf{vis}_C\ c \\
\|[\ \textbf{vis}_B\ b_\wedge; \textbf{vis}_C\ c_\wedge \\
\quad \textbf{vis}_B\ b_\vee; \textbf{vis}_C\ c_\vee \\
\quad \textbf{vis}_A\ a_b, a_c
\end{array}$$

Macro-atomicity

Let «*P*» mean “program fragment *P* executed atomically: ephemeral visibles not seen, nor interior control flow.”

Macro-atomicity

Let «*P*» mean "program fragment *P* executed atomically: ephemeral visibles not seen, nor interior control flow."

1. For classical primitive statement S we have «S» = S.
   This is by definition of primitive statements' semantics.

Macro-atomicity

Let «P» mean "program fragment P executed atomically: ephemeral visibles not seen, nor interior control flow."

1. For classical primitive statement S we have «S» = S.
   This is by definition of primitive statements' semantics.

2. For fragments P,Q we have «P»;«Q» $\sqsubseteq$ «P;Q».
   This is by definition of «•».
   Informally, it is because the ephemerals are visible at left but hidden at right.

Macro-atomicity

Let «P» mean "program fragment P executed atomically: ephemeral visibles not seen, nor interior control flow."

1. For classical primitive statement S we have «S» = S.
2. For fragments P,Q we have «P»;«Q» $\sqsubseteq$ «P;Q».

3. For fragments P,Q we have «P»;«Q» = «P;Q» provided the intermediate visibles can be inferred from the external visibles.

   Informally, it is because the intermediates' being inferred means that hiding them has no effect.

Macro-atomicity

Let «P» mean "program fragment P executed atomically: ephemeral visibles not seen, nor interior control flow."

1. For classical primitive statement S we have «S» = S.
2. For fragments P,Q we have «P»;«Q» $\sqsubseteq$ «P;Q».
3. For fragments P,Q we have «P»;«Q» = «P;Q» provided the intermediate visibles can be inferred from the external visibles.

4. For fragments P,Q we have $P = Q$ implies «P» = «Q». Trivial.

Macro-atomicity

Let «P» mean "program fragment P executed atomically: ephemeral visibles not seen, nor interior control flow."

1. For classical primitive statement S we have «S» = S.
2. For fragments P,Q we have «P»;«Q» $\sqsubseteq$ «$P;Q$».
3. For fragments P,Q we have «P»;«Q» = «$P;Q$» provided the intermediate visibles can be inferred from the external visibles.
4. For fragments P,Q we have $P = Q$ implies «P» = «Q».

If two straight-line primitive sequences $S_1;S_2;\ldots;S_M$ and $T_1;T_2;\ldots;T_N$ are equal classically, and no visible is assigned-to twice at right, then the sequences are security-equal as well.

Appendix

- Derivation of the Encryption Lemma
- Derivation of The Three Judges
- Derivation of Oblivious Transfer

Oblivious transfer

$$\mathbf{vis}_A\ a, a_b$$

$$\mathbf{vis}_B\ b_\vee, b_\wedge$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

Oblivious transfer

$\mathbf{vis}_A\ a, a_b$
$\mathbf{vis}_B\ b_\vee, b_\wedge$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

Oblivious transfer

$$\textbf{vis}_A \; a, a_b$$
$$\textbf{vis}_B \; b_\vee, b_\wedge$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

Oblivious transfer

$$\begin{array}{l}\mathbf{vis}_A\ a, a_b \\ \mathbf{vis}_B\ b_\vee, b_\wedge\end{array}$$

$$\begin{array}{l}|[\quad \mathbf{vis}\ x; \mathbf{vis}_A\ a' \\ \quad a' :\in \{0,1\} \\ \quad x := a \oplus a' \\ ]| \\ a_b := (b_\vee \lhd a \rhd b_\wedge)\end{array}$$

Encryption Lemma

Oblivious transfer

$$\mathbf{vis}_A\ a, a_b$$
$$\mathbf{vis}_B\ b_\vee, b_\wedge$$

$$\begin{array}{l} |[\quad \mathbf{vis}\ x; \mathbf{vis}_A\ a' \\ \quad a' :\in \{0,1\} \\ \quad x := a \oplus a' \\ ]| \end{array}$$

Encryption Lemma

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

Oblivious transfer

$$a' :\in \{0, 1\}$$
$$x := a \oplus a'$$

$\mathbf{vis}_A\ a, a_b$
$\mathbf{vis}_B\ b_\vee, b_\wedge$
$|[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a'$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

Oblivious transfer

$$\begin{array}{l} a' :\in \{0,1\} \\ x := a \oplus a' \\ |[\quad \mathbf{vis}\ y_\vee, y_\wedge \\ \quad\quad \mathbf{vis}_B\ b'_0, b'_1 \\ \quad\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\} \\ \quad\quad y_\vee := b_\vee \oplus b'_{\neg x} \\ \quad\quad y_\wedge := b_\wedge \oplus b'_x \quad]| \\ a_b := (b_\vee \lhd a \rhd b_\wedge) \end{array}$$

$$\begin{array}{l} \mathbf{vis}_A\ a, a_b \\ \mathbf{vis}_B\ b_\vee, b_\wedge \\ |[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a' \end{array}$$

Encryption Lemma twice

Oblivious transfer

$$
\begin{aligned}
& a' :\in \{0,1\} \\
& x := a \oplus a' \\
& |[\quad \mathbf{vis}\ y_\vee, y_\wedge \\
& \qquad \mathbf{vis}_B\ b'_0, b'_1 \\
& \qquad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\} \\
& \qquad y_\vee := b_\vee \oplus b'_{\neg x} \\
& \qquad y_\wedge := b_\wedge \oplus b'_x \quad]| \\
& a_b := (b_\vee \lhd a \rhd b_\wedge)
\end{aligned}
$$

$$
\begin{aligned}
& \mathbf{vis}_A\ a, a_b \\
& \mathbf{vis}_B\ b_\vee, b_\wedge \\
& |[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a'
\end{aligned}
$$

Encryption Lemma twice

Oblivious transfer

$$a' :\in \{0,1\}$$
$$x := a \oplus a'$$

$$b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

$\mathbf{vis}_A\ a, a_b$
$\mathbf{vis}_B\ b_\vee, b_\wedge$
$|[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a'$
$\quad \mathbf{vis}\ y_\vee, y_\wedge$
$\quad \mathbf{vis}_B\ b'_0, b'_1$

Oblivious transfer

$$a' :\in \{0,1\}$$

$$x \leftarrow a \oplus a'$$

EL

$$b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}$$

$$y_\vee \leftarrow b_\vee \oplus b'_{\neg x}$$

$$y_\wedge \leftarrow b_\wedge \oplus b'_x$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

$\mathbf{vis}_A\ a, a_b$
$\mathbf{vis}_B\ b_\vee, b_\wedge$
$|[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a'$
$\quad \mathbf{vis}\ y_\vee, y_\wedge$
$\quad \mathbf{vis}_B\ b'_0, b'_1$

Oblivious transfer

$$x := a \oplus a'$$

$$y_\vee := b_\vee \oplus b'_{\neg x}$$

$$y_\wedge := b_\wedge \oplus b'_x$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

$\mathbf{vis}_A\ a, a_b$
$\mathbf{vis}_B\ b_\vee, b_\wedge$
$|[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a'$
$\quad \mathbf{vis}\ y_\vee, y_\wedge$
$\quad \mathbf{vis}_B\ b'_0, b'_1$
$\quad a' :\in \{0,1\}$
$\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}$

Oblivious transfer

$$x := a \oplus a'$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$

$$\begin{array}{l}
\mathbf{vis}_A\ a, a_b \\
\mathbf{vis}_B\ b_\vee, b_\wedge \\
|[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a' \\
\quad \mathbf{vis}\ y_\vee, y_\wedge \\
\quad \mathbf{vis}_B\ b'_0, b'_1 \\
\quad a' :\in \{0,1\} \\
\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}
\end{array}$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

All three variables are visible to everyone; but they learn nothing at all from them.

Oblivious transfer

$$x := a \oplus a'$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$

$$\begin{array}{l}
\textbf{vis}_A\ a, a_b \\
\textbf{vis}_B\ b_\vee, b_\wedge \\
|[\ \textbf{vis}\ x; \textbf{vis}_A\ a' \\
\quad \textbf{vis}\ y_\vee, y_\wedge \\
\quad \textbf{vis}_B\ b'_0, b'_1 \\
\quad a' :\in \{0,1\} \\
\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}
\end{array}$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

All three variables are visible to everyone; but they learn nothing at all from them.

And yet...

Oblivious transfer

$$\begin{aligned}
&\mathbf{vis}_A\ a, a_b \\
&\mathbf{vis}_B\ b_\vee, b_\wedge \\
&|[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a' \\
&\quad \mathbf{vis}\ y_\vee, y_\wedge \\
&\quad \mathbf{vis}_B\ b'_0, b'_1 \\
&\quad a' :\in \{0,1\} \\
&\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}
\end{aligned}$$

$$x := a \oplus a'$$

$$y_\vee := b_\vee \oplus b'_{\neg x}$$

$$y_\wedge := b_\wedge \oplus b'_x$$

$$a_b := (b_\vee \lhd a \rhd b_\wedge)$$

Oblivious transfer

$$\begin{aligned}
&\textbf{vis}_A\ a, a_b\\
&\textbf{vis}_B\ b_\vee, b_\wedge\\
&|[\ \textbf{vis}\ x; \textbf{vis}_A\ a'\\
&\quad \textbf{vis}\ y_\vee, y_\wedge\\
&\quad \textbf{vis}_B\ b'_0, b'_1\\
&\quad a' :\in \{0,1\}\\
&\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}
\end{aligned}$$

$$x := a \oplus a'$$

$$y_\vee := b_\vee \oplus b'_{\neg x}$$

$$y_\wedge := b_\wedge \oplus b'_x$$

$$a_b := (y_\vee \oplus b'_{\neg x} \ \lhd a \rhd\ y_\wedge \oplus b'_x)$$

Oblivious transfer

$$x := a \oplus a'$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$

$$\begin{array}{l}
\textbf{vis}_A\ a, a_b \\
\textbf{vis}_B\ b_\vee, b_\wedge \\
|[\ \textbf{vis}\ x; \textbf{vis}_A\ a' \\
\quad \textbf{vis}\ y_\vee, y_\wedge \\
\quad \textbf{vis}_B\ b'_0, b'_1 \\
\quad a' :\in \{0,1\} \\
\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}
\end{array}$$

$$a_b := (y_\vee \oplus b'_{a'} \ \lhd a \rhd \ y_\wedge \oplus b'_{a'})$$

Oblivious transfer

$$x := a \oplus a'$$

$$y_\vee := b_\vee \oplus b'_{\neg x}$$

$$y_\wedge := b_\wedge \oplus b'_x$$

$$\begin{array}{l}
\textbf{vis}_A\ a, a_b \\
\textbf{vis}_B\ b_\vee, b_\wedge \\
|[\ \textbf{vis}\ x; \textbf{vis}_A\ a' \\
\quad \textbf{vis}\ y_\vee, y_\wedge \\
\quad \textbf{vis}_B\ b'_0, b'_1 \\
\quad a' :\in \{0,1\} \\
\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}
\end{array}$$

$$a_b := (y_\vee \lhd a \rhd y_\wedge) \oplus b'_{a'}$$

Oblivious transfer

$$\mathbf{vis}_A\ a, a_b$$
$$\mathbf{vis}_B\ b_\vee, b_\wedge$$
$$|[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a'$$
$$\mathbf{vis}\ y_\vee, y_\wedge$$
$$\mathbf{vis}_B\ b'_0, b'_1$$
$$a' :\in \{0,1\}$$
$$b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}$$

$$x := a \oplus a'$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$

$$\underline{a_b} := \underline{(y_\vee \lhd a \rhd y_\wedge)} \oplus b'_{a'}$$

Both visible to A

Oblivious transfer

$$x := a \oplus a'$$

$$y_\vee := b_\vee \oplus b'_{\neg x}$$

$$y_\wedge := b_\wedge \oplus b'_x$$

$$a'' := b'_{a'}$$

$$a_b := (y_\vee \lhd a \rhd y_\wedge) \oplus a''$$

$\mathbf{vis}_A\ a, a_b$
$\mathbf{vis}_B\ b_\vee, b_\wedge$
$|[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a'$
$\quad \mathbf{vis}\ y_\vee, y_\wedge$
$\quad \mathbf{vis}_B\ b'_0, b'_1$
$\quad a' :\in \{0,1\}$
$\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}$
$\quad \mathbf{vis}_A\ a''$

Oblivious transfer

$$x := a \oplus a'$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$

$$a'' := (b'_1 \lhd a' \rhd b'_0)$$

$$a_b := (y_\vee \lhd a \rhd y_\wedge) \oplus a''$$

$$
\begin{array}{l}
\textbf{vis}_A\ a, a_b \\
\textbf{vis}_B\ b_\vee, b_\wedge \\
|[\ \textbf{vis}\ x; \textbf{vis}_A\ a' \\
\quad \textbf{vis}\ y_\vee, y_\wedge \\
\quad \textbf{vis}_B\ b'_0, b'_1 \\
\quad a' :\in \{0,1\} \\
\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\} \\
\quad \textbf{vis}_A\ a''
\end{array}
$$

Oblivious transfer

$$a'' := (b'_1 \lhd a' \rhd b'_0)$$

$$x := a \oplus a'$$

$$y_\vee := b_\vee \oplus b'_{\neg x}$$

$$y_\wedge := b_\wedge \oplus b'_x$$

$$a_b := (y_\vee \lhd a \rhd y_\wedge) \oplus a''$$

$\mathbf{vis}_A\ a, a_b$
$\mathbf{vis}_B\ b_\vee, b_\wedge$
$|[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a'$
$\quad \mathbf{vis}\ y_\vee, y_\wedge$
$\quad \mathbf{vis}_B\ b'_0, b'_1$
$\quad a' :\in \{0,1\}$
$\quad b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}$
$\quad \mathbf{vis}_A\ a''$

Oblivious transfer

$$a' :\in \{0,1\}$$
$$b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}$$
$$a'' := (b'_1 \lhd a' \rhd b'_0)$$

$$\begin{array}{l}
\textbf{vis}_A\ a, a_b \\
\textbf{vis}_B\ b_\vee, b_\wedge \\
|[\ \textbf{vis}\ x; \textbf{vis}_A\ a' \\
\quad \textbf{vis}\ y_\vee, y_\wedge \\
\quad \textbf{vis}_B\ b'_0, b'_1 \\
\quad \textbf{vis}_A\ a''
\end{array}$$

$$x := a \oplus a'$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$
$$a_b := (y_\vee \lhd a \rhd y_\wedge) \oplus a''$$

Oblivious transfer

$$a' :\in \{0,1\}$$
$$b'_0 :\in \{0,1\};\, b'_1 :\in \{0,1\}$$
$$a'' := (b'_1 \lhd a' \rhd b'_0)$$

$$\begin{array}{l} \mathbf{vis}_A\ a, a_b \\ \mathbf{vis}_B\ b_\vee, b_\wedge \\ |[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a' \\ \quad \mathbf{vis}\ y_\vee, y_\wedge \\ \quad \mathbf{vis}_B\ b'_0, b'_1 \\ \quad \mathbf{vis}_A\ a'' \end{array}$$

$$x := a \oplus a'$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$
$$a_b := (y_\vee \lhd a \rhd y_\wedge) \oplus a''$$

Done in advance, by a trusted third party

At first, prepare...

$$a' :\in \{0,1\}$$
$$b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}$$
$$a'' := (b'_1 \lhd a' \rhd b'_0)$$

$$\begin{array}{l} \mathbf{vis}_A\ a, a_b \\ \mathbf{vis}_B\ b_\vee, b_\wedge \\ |[\ \mathbf{vis}\ x; \mathbf{vis}_A\ a' \\ \quad \mathbf{vis}\ y_\vee, y_\wedge \\ \quad \mathbf{vis}_B\ b'_0, b'_1 \\ \quad \mathbf{vis}_A\ a'' \end{array}$$

$$x := a \oplus a'$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$
$$a_b := (y_\vee \lhd a \rhd y_\wedge) \oplus a''$$

The actual transfer

$$a' :\in \{0,1\}$$
$$b'_0 :\in \{0,1\}; b'_1 :\in \{0,1\}$$
$$a'' := (b'_1 \lhd a' \rhd b'_0)$$

$\mathbf{vis}_A\ a, a_b$
$\mathbf{vis}_B\ b_\vee, b_\wedge$

$|[\ \mathbf{vis}_A\ a', a''$
$\quad \mathbf{vis}_B\ b'_0, b'_1$

$\quad \mathbf{vis}\ x, y_\vee, y_\wedge$

Later...

$$x := a \oplus a'$$
$$y_\vee := b_\vee \oplus b'_{\neg x}$$
$$y_\wedge := b_\wedge \oplus b'_x$$
$$a_b := (y_\vee \lhd a \rhd y_\wedge) \oplus a''$$

Appendix

- Derivation of the Encryption Lemma
- Derivation of The Three Judges
- Derivation of Oblivious Transfer