# A Formal Analysis of the Norwegian E-voting Protocol

Véronique Cortier & Cyrille Wiedling

LORIA - CNRS, Nancy, France

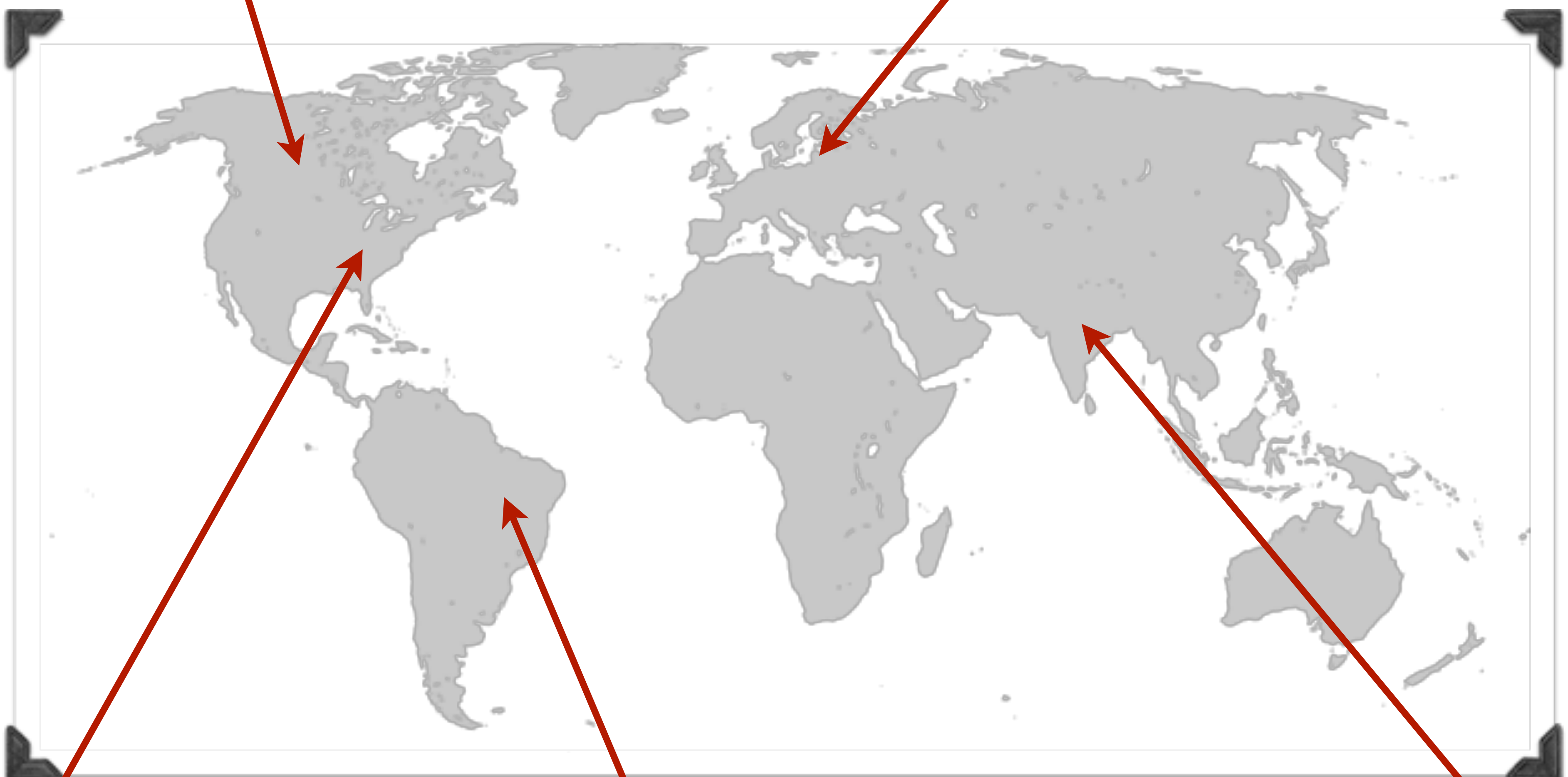
March 26th, 2012

Project supported by the
European Research Council

# E-voting : a worldwide expansion

**Canada** : Since 2004 at the Provincial level. (EVM and (later) Internet voting.)

**Estonia** : 2005, first legally binding vote using Internet.



**USA** : EVM used for legally binding vote since 1996.

**India** : legally binding e-voting with EVM since 2002.

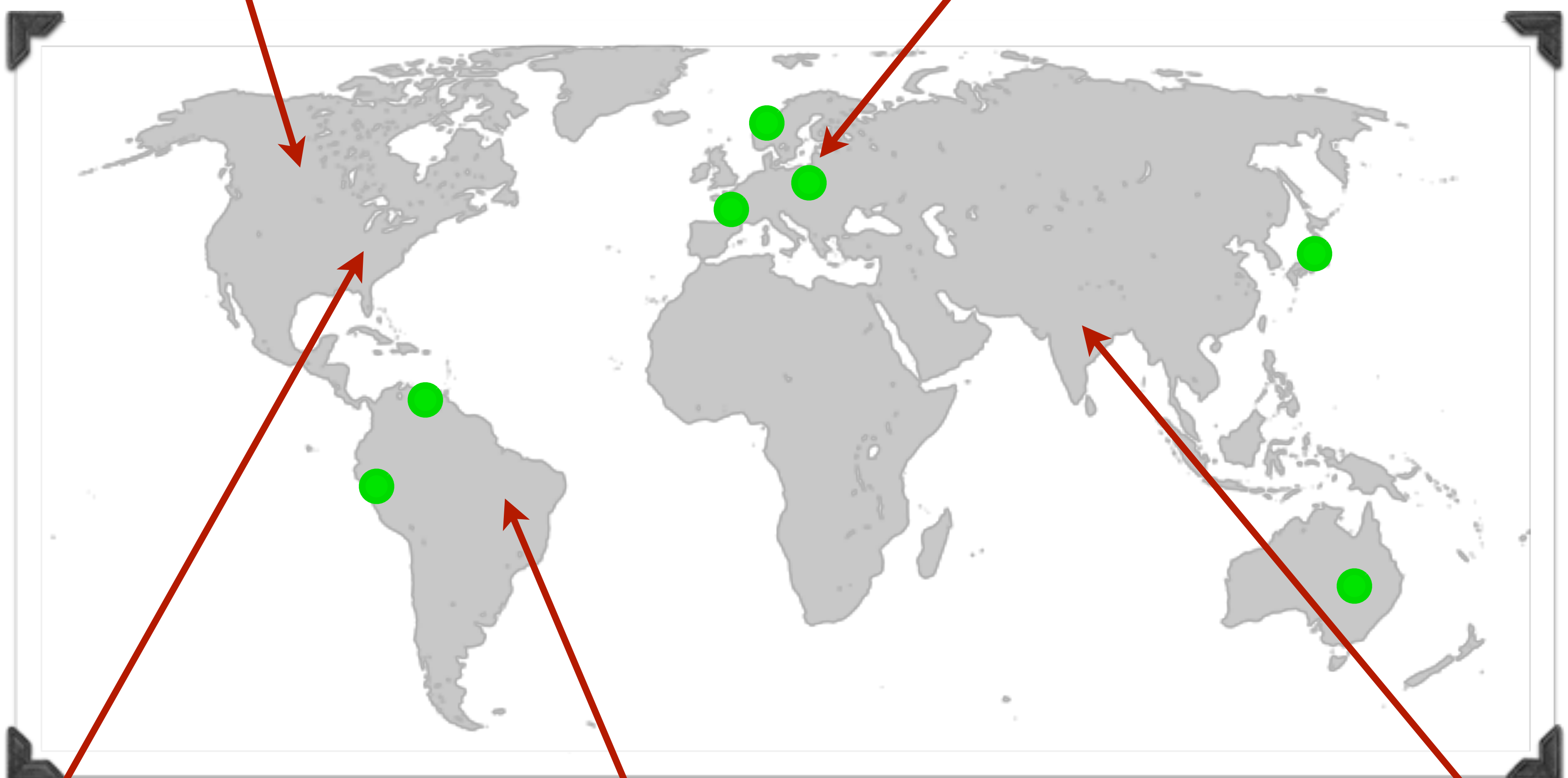
**Brazil** : legally binding e-vote with EVM since 2000.

# E-voting : a worldwide expansion

**Canada** : Since 2004 at the Provincial level. (EVM and (later) Internet voting.)

**Estonia** : 2005, first legally binding vote using Internet.

But also :
Norway
France,
Poland,
...



**USA** : EVM used for legally binding vote since 1996.

**India** : legally binding e-voting with EVM since 2002.

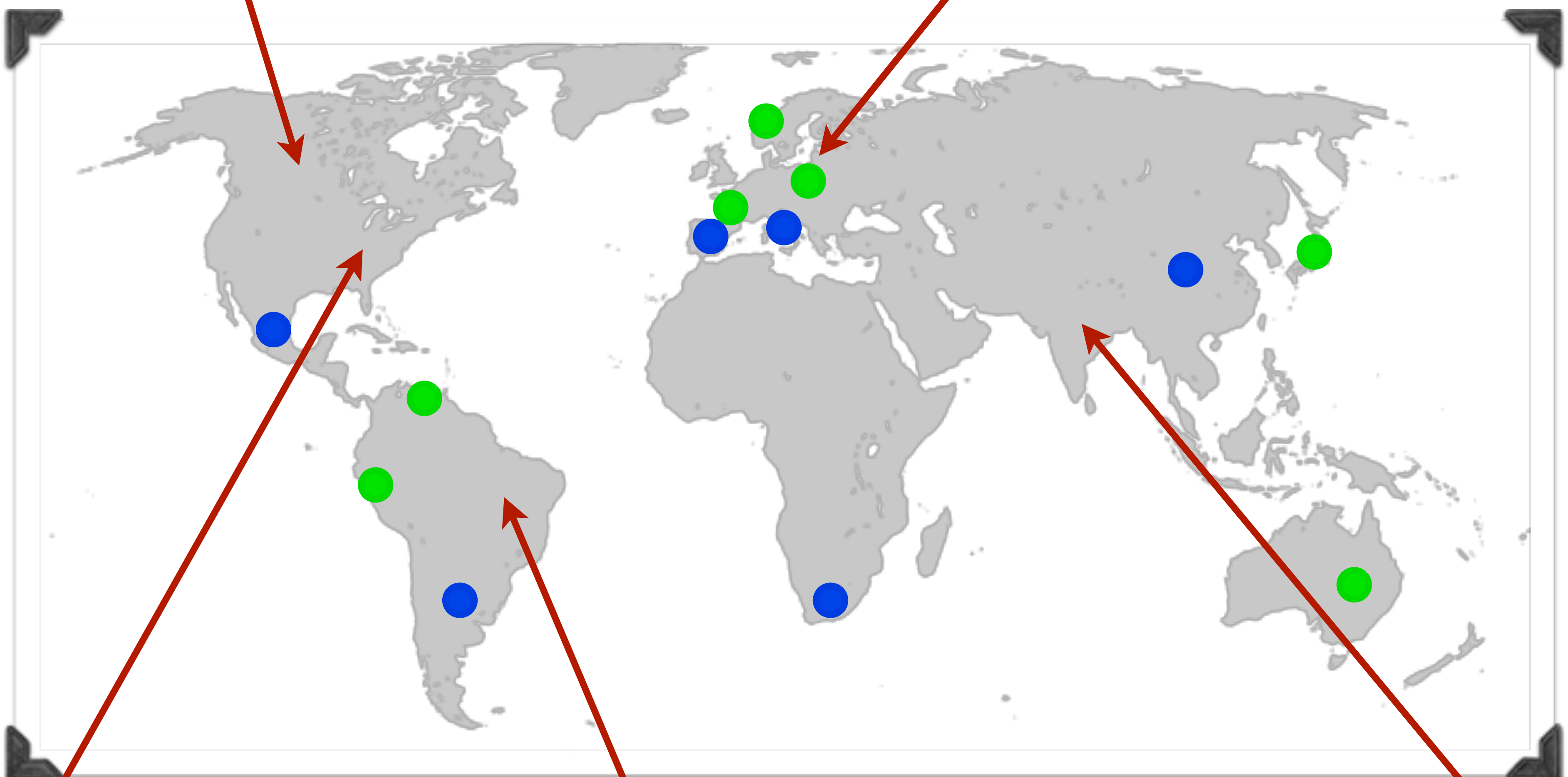
**Brazil** : legally binding e-vote with EVM since 2000.

# E-voting : a worldwide expansion

**Canada** : Since 2004 at the Provincial level. (EVM and (later) Internet voting.)

**Estonia** : 2005, first legally binding vote using Internet.

But also :
Norway
France,
Poland,
...



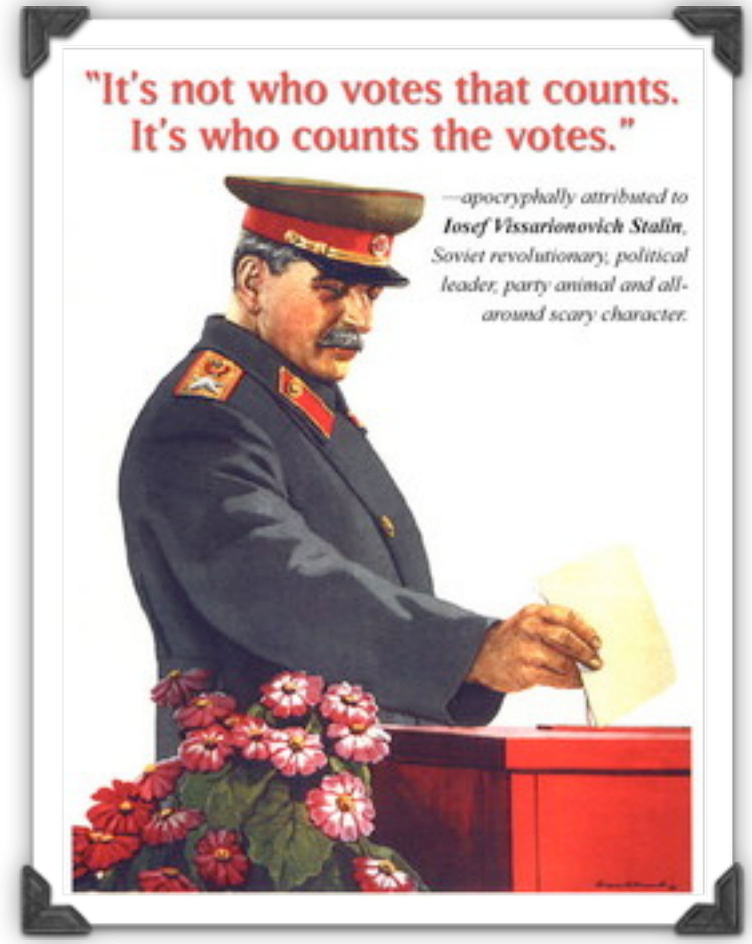
Planning in :
Mexico,
China,
Spain,
...

**USA** : EVM used for legally binding vote since 1996.

**India** : legally binding e-voting with EVM since 2002.

**Brazil** : legally binding e-vote with EVM since 2000.

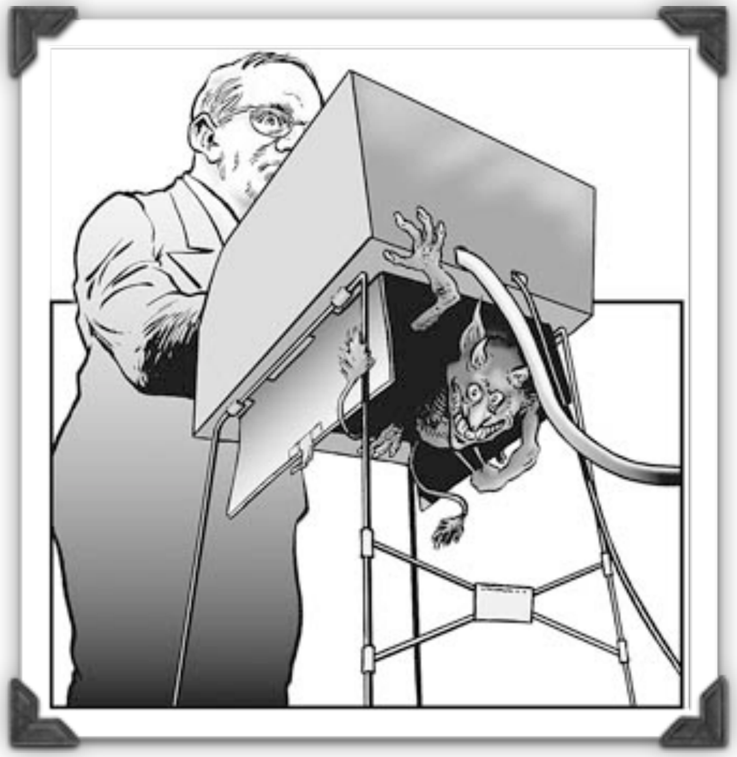
# Why using E-voting ?




**Efficiency** and **Reliability**
in collecting and tallying votes
(less Human errors/cheating in counting)

**Convenient** way of voting
Possibility of voting from home
or anywhere else.
(More people may vote)

# E-voting is not a wonderland...



Systems may be **vulnerable to attacks** :

- Diebold Machines in the U.S. (Candice Hoke, 2008)
- Paperless EVM in India. (A. Halderman, R. Gonggrijp, 2010)

Some countries just decide to **stop E-voting** :

- Germany
- Ireland
- United Kingdom

# A powerful attacker

Presence of an **attacker** who :

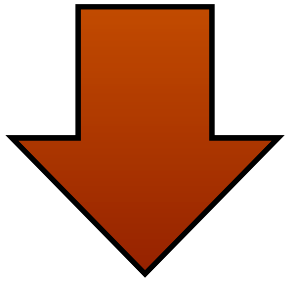
- can **read** every message sent on the network,
- can **intercept** messages,
- can **create** and **send** new messages.
- can **vote** himself.

# A powerful attacker

Presence of an **attacker** who :

- can **read** every message sent on the network,
- can **intercept** messages,
- can **create** and **send** new messages.
- can **vote** himself.

Powerful attacker



There is a crucial need to verify protocols before using them !

# Contributions

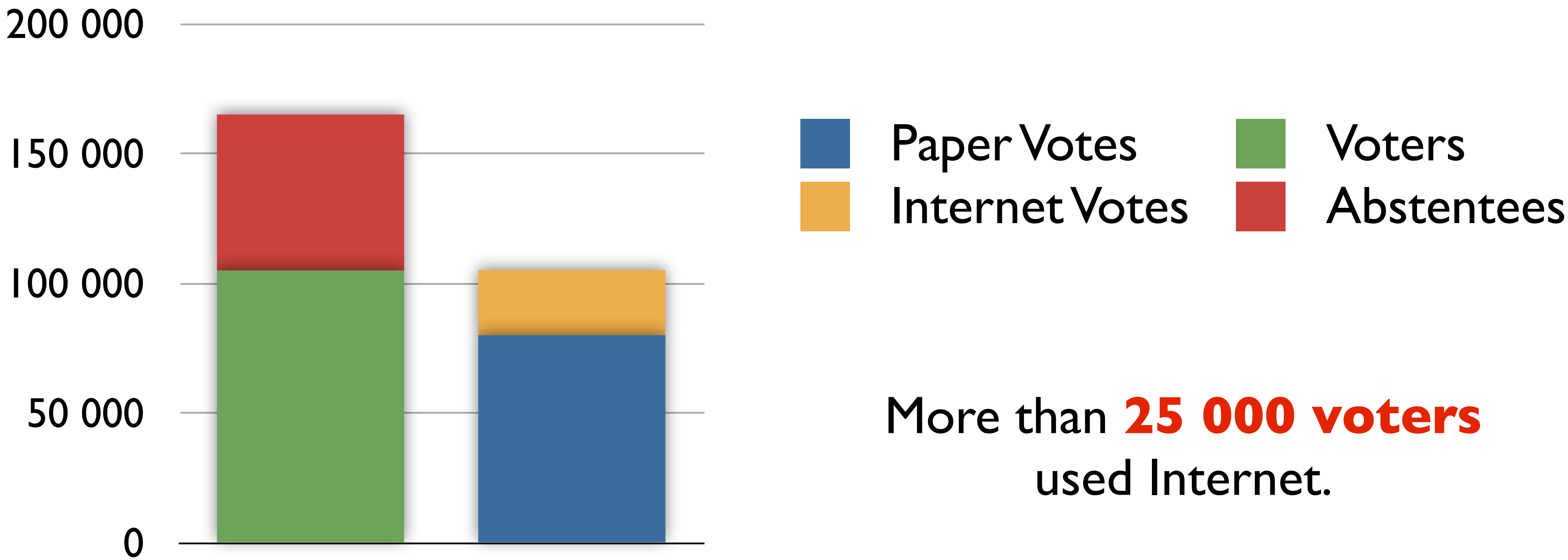
- **Modeling** of an implemented and tested protocol,
    - modeling of **complex primitives**,
    - modeling of **trust assumptions**.

- Analysis of the property of **vote-privacy**,

- Using of **ProVerif** tool over a simple modeling to explore further cases of corruption.

# The Norwegian E-voting protocol

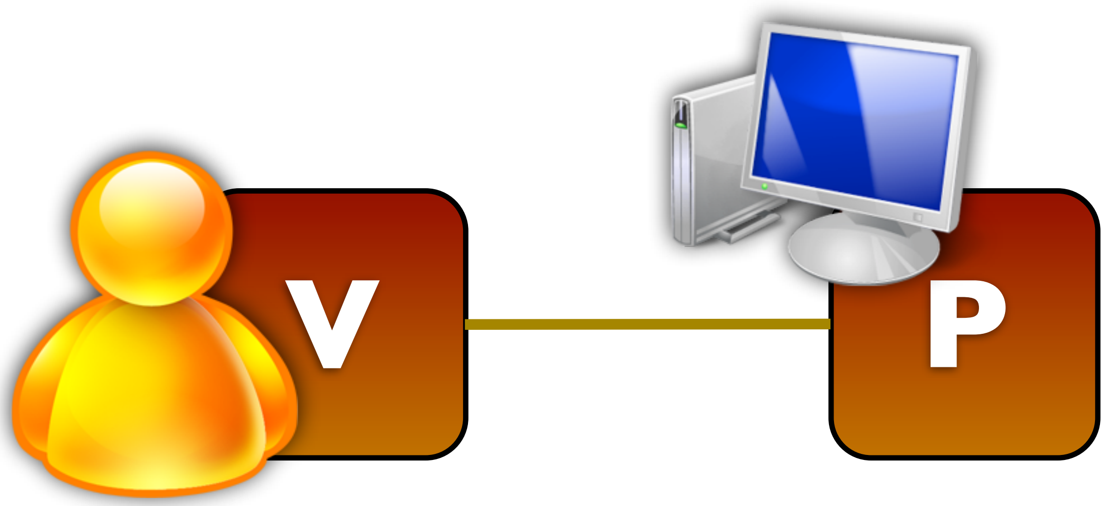
- Developed by **ErgoGroup**,
- Used in **municipal** and **county** elections,
- Already implemented and tested in **real conditions**,



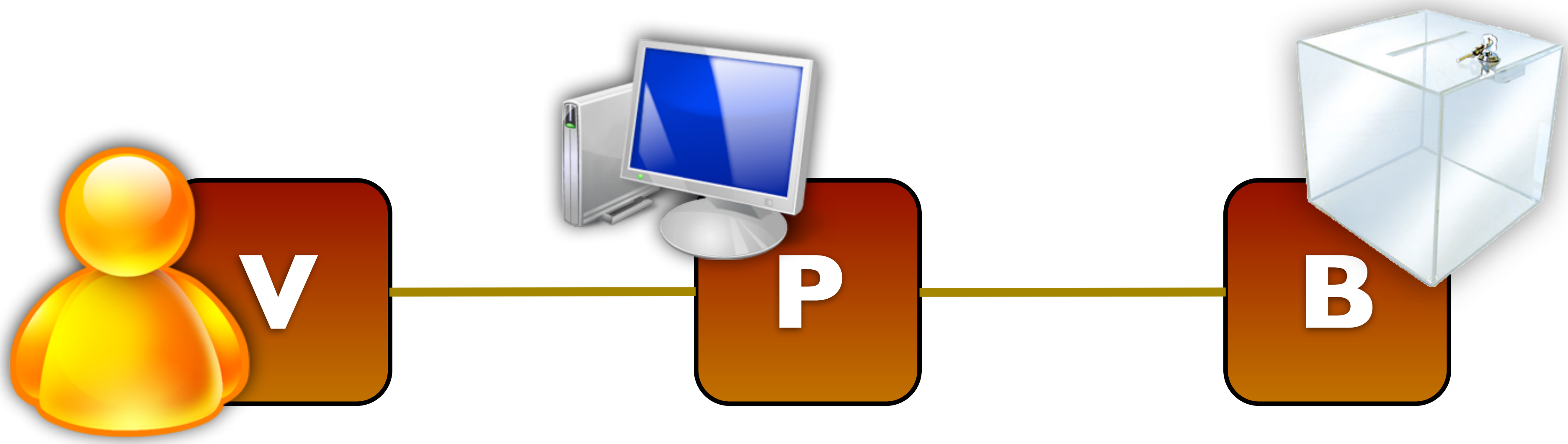

**2011 elections results in the 10 participating cities**

More than **25 000 voters** used Internet.

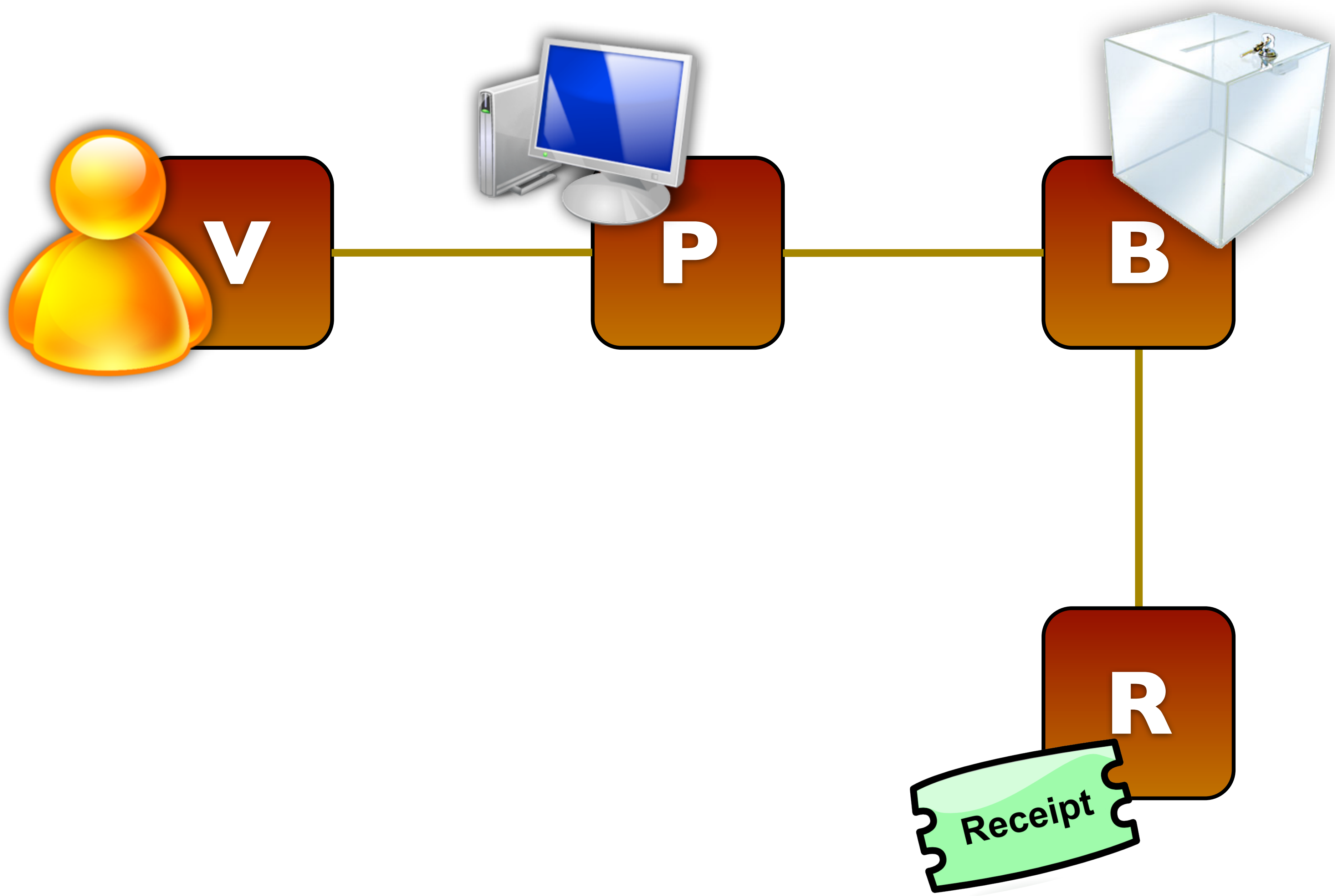
# Players of the protocol



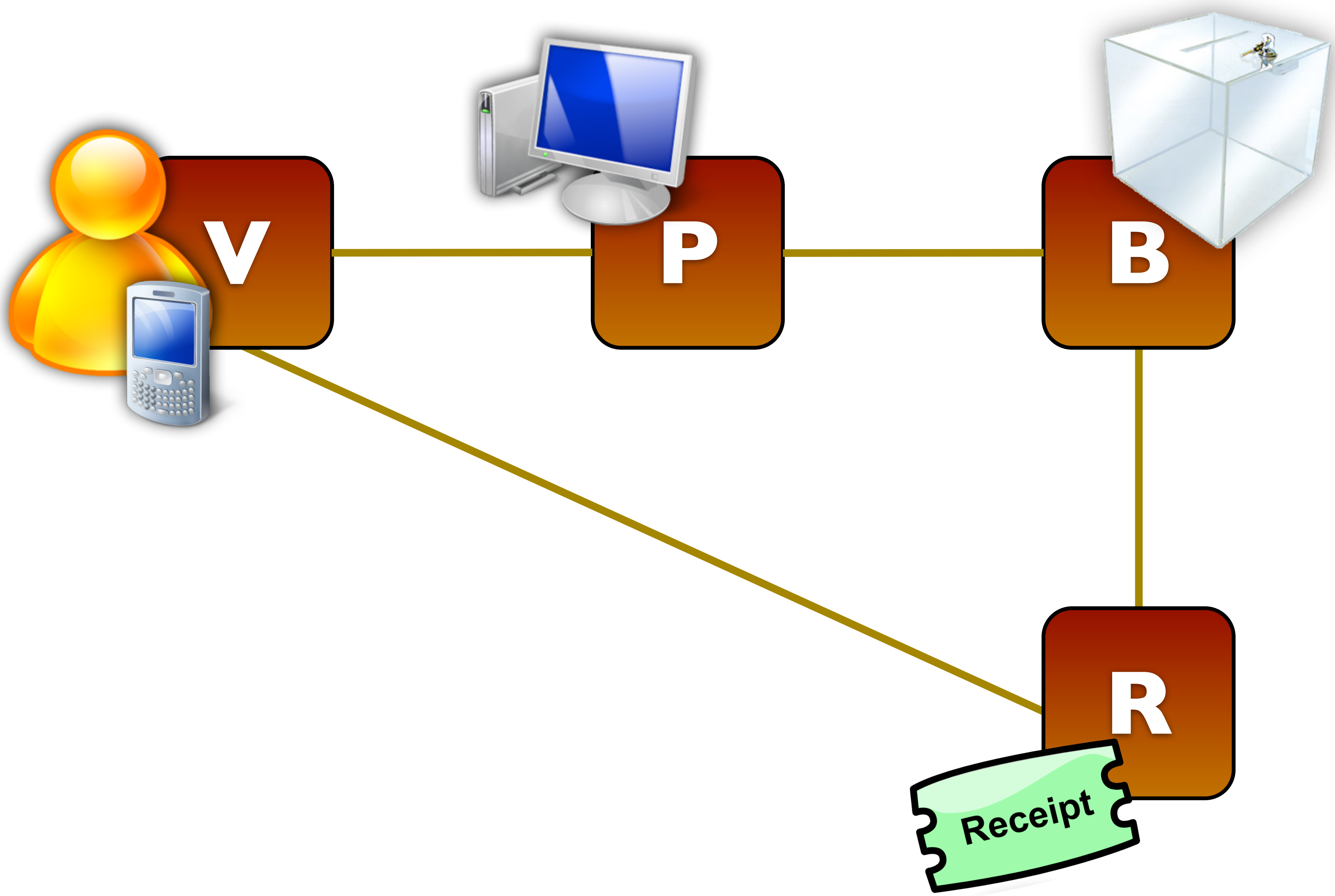
# Players of the protocol



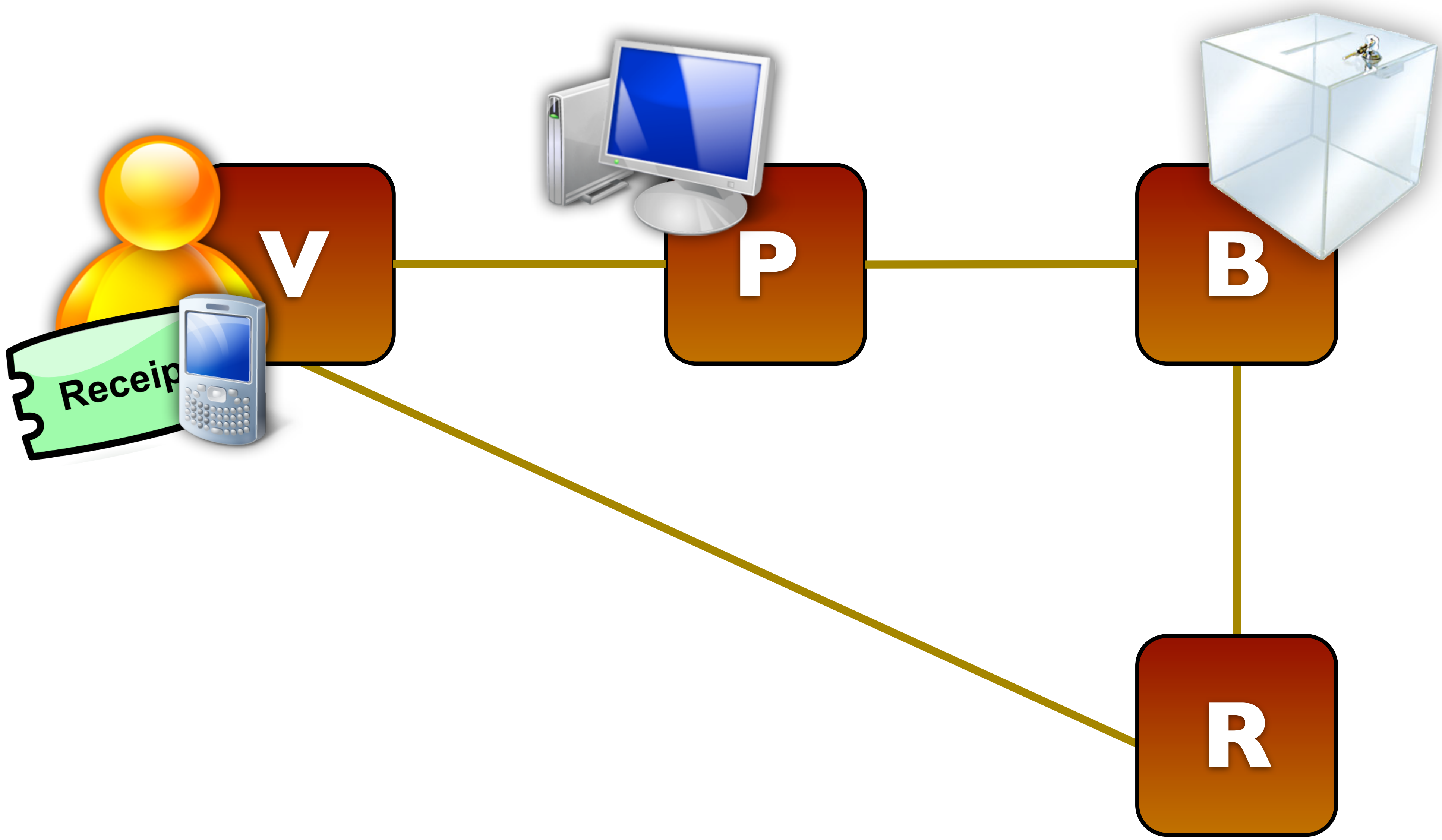
# Players of the protocol



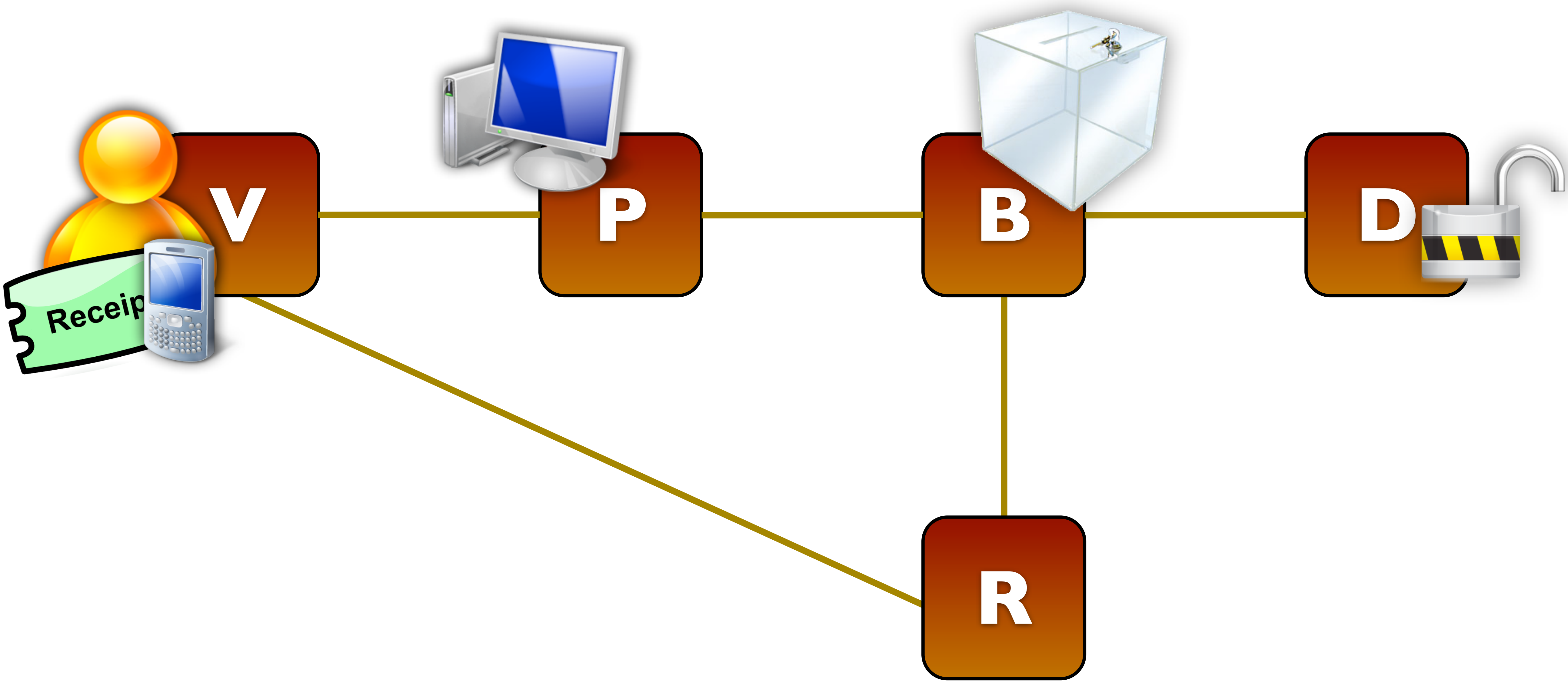
# Players of the protocol



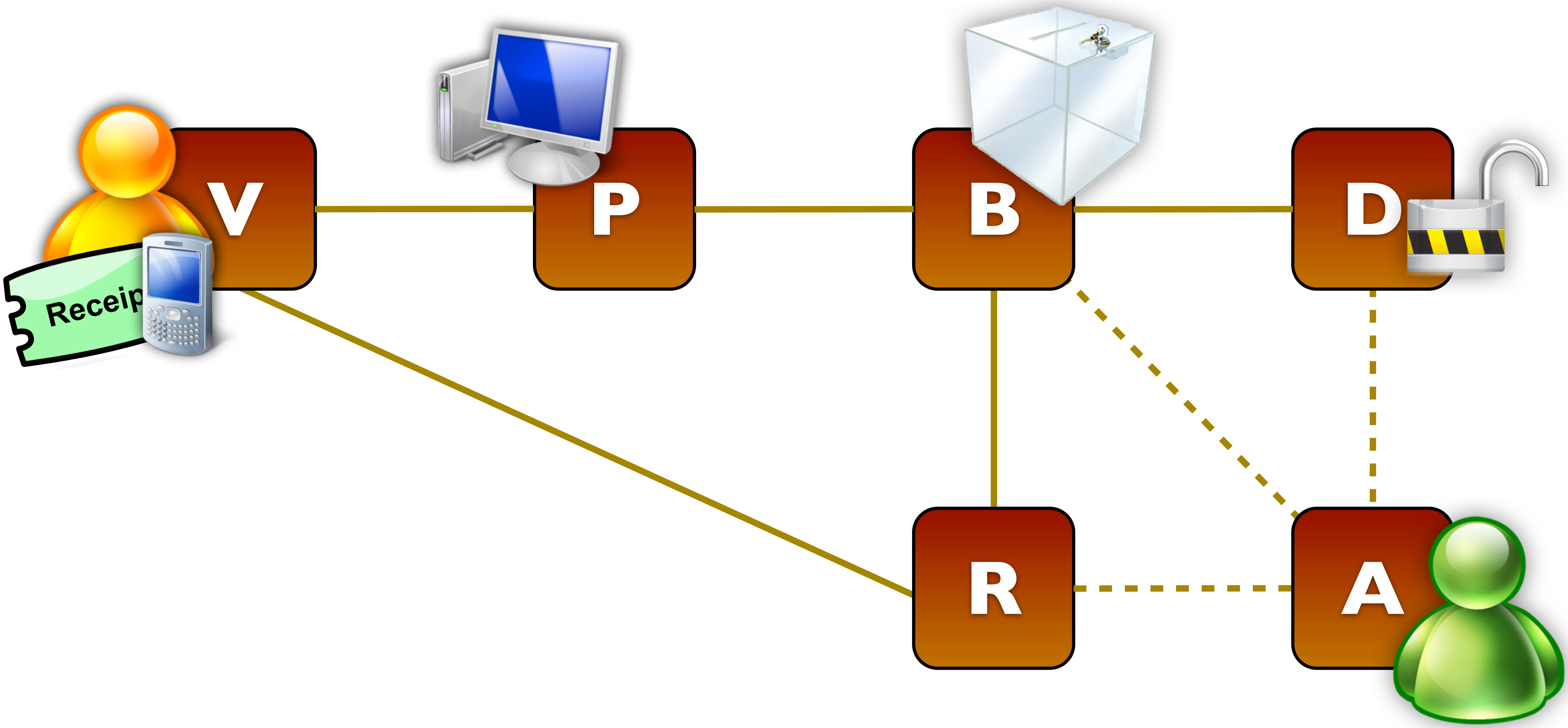
# Players of the protocol



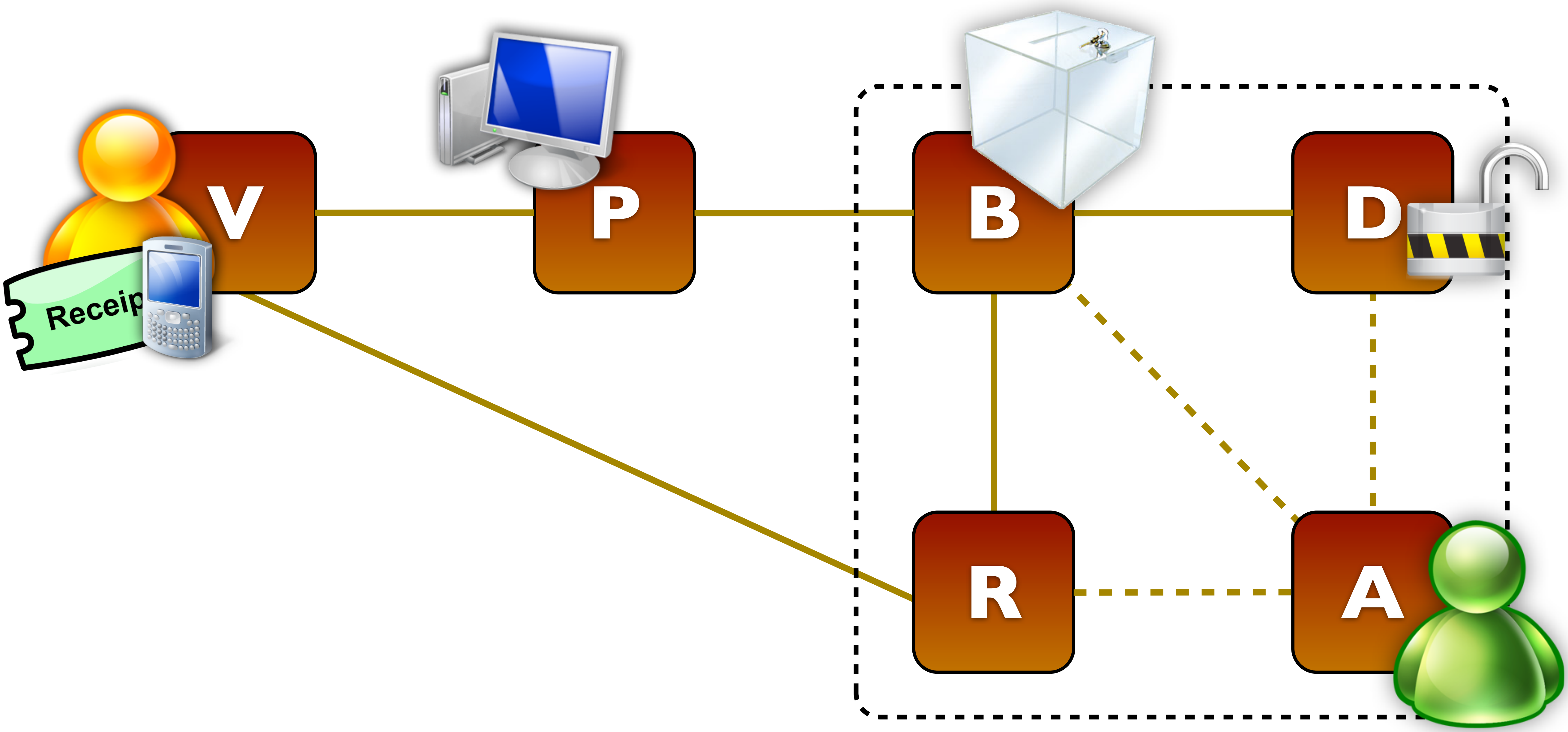
# Players of the protocol



# Players of the protocol

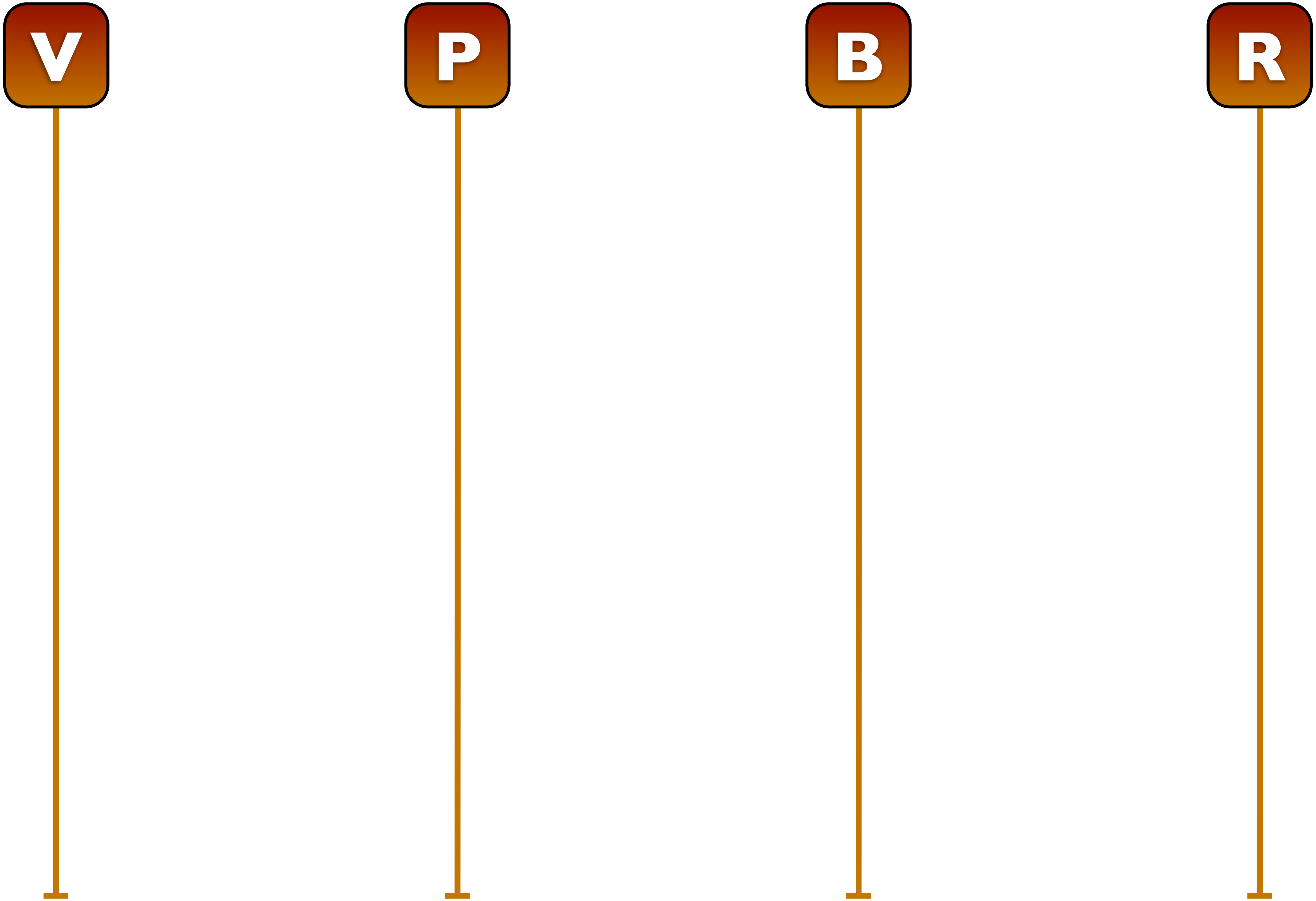


# Players of the protocol

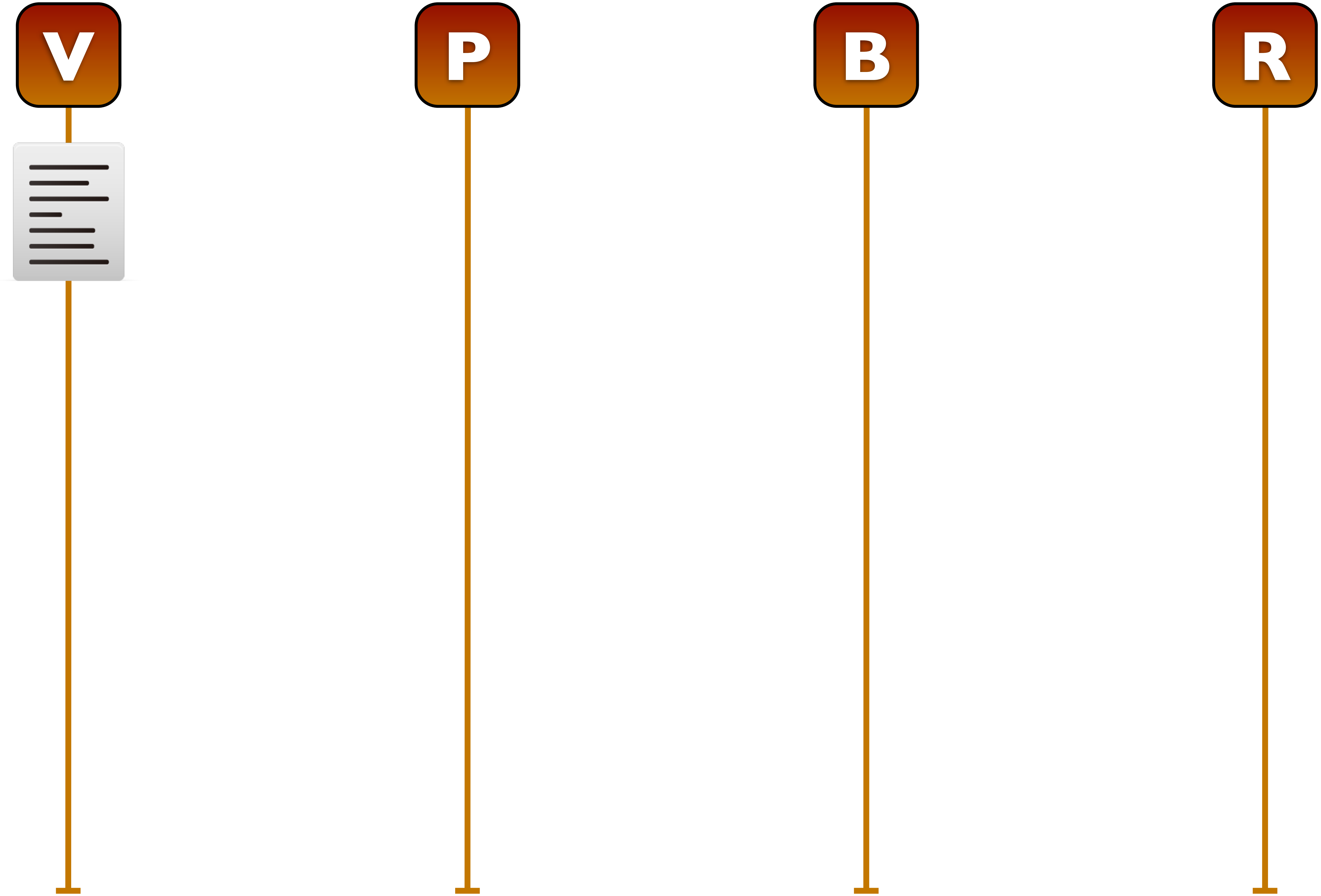



**Infrastructure players**

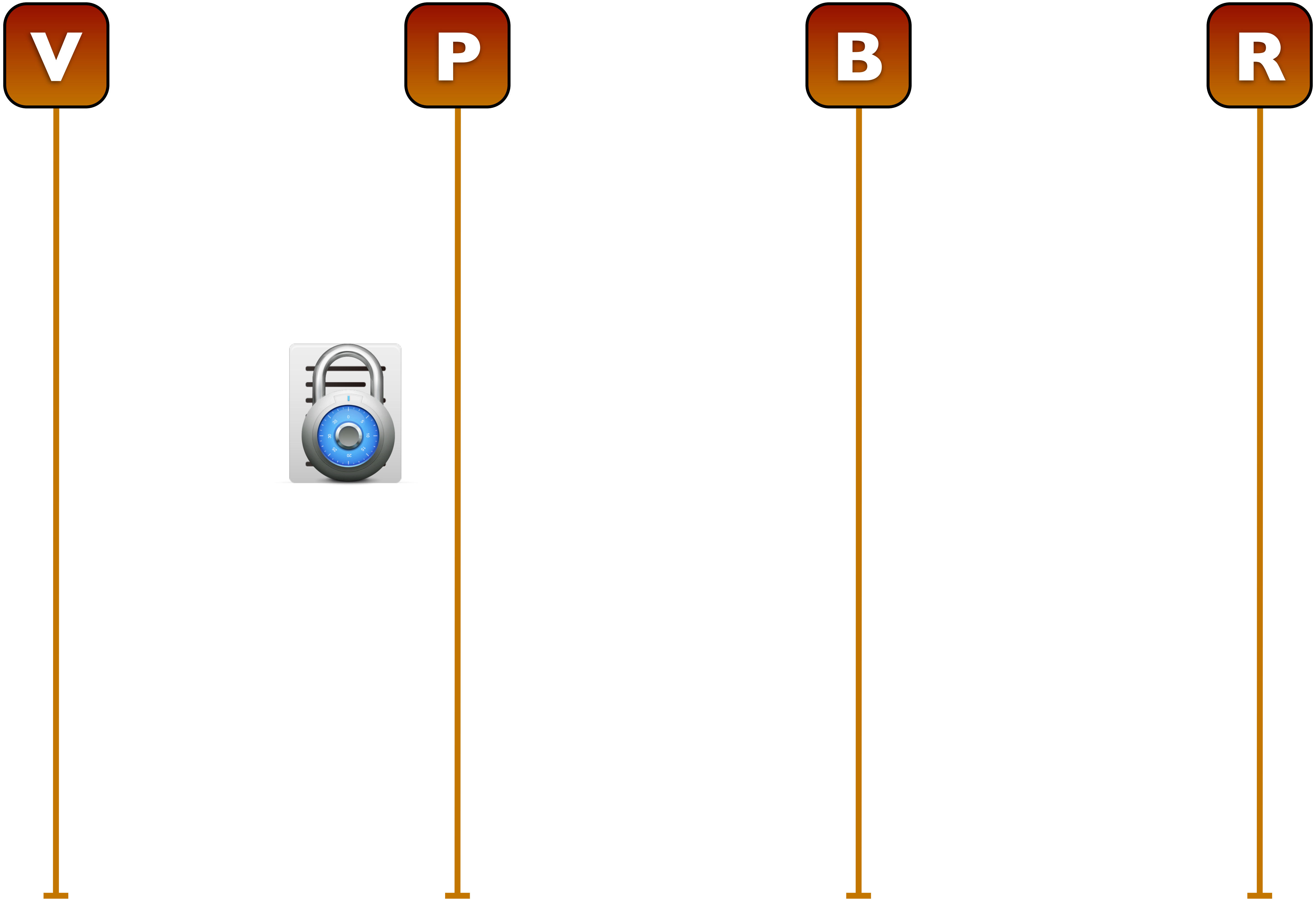
# Submission process



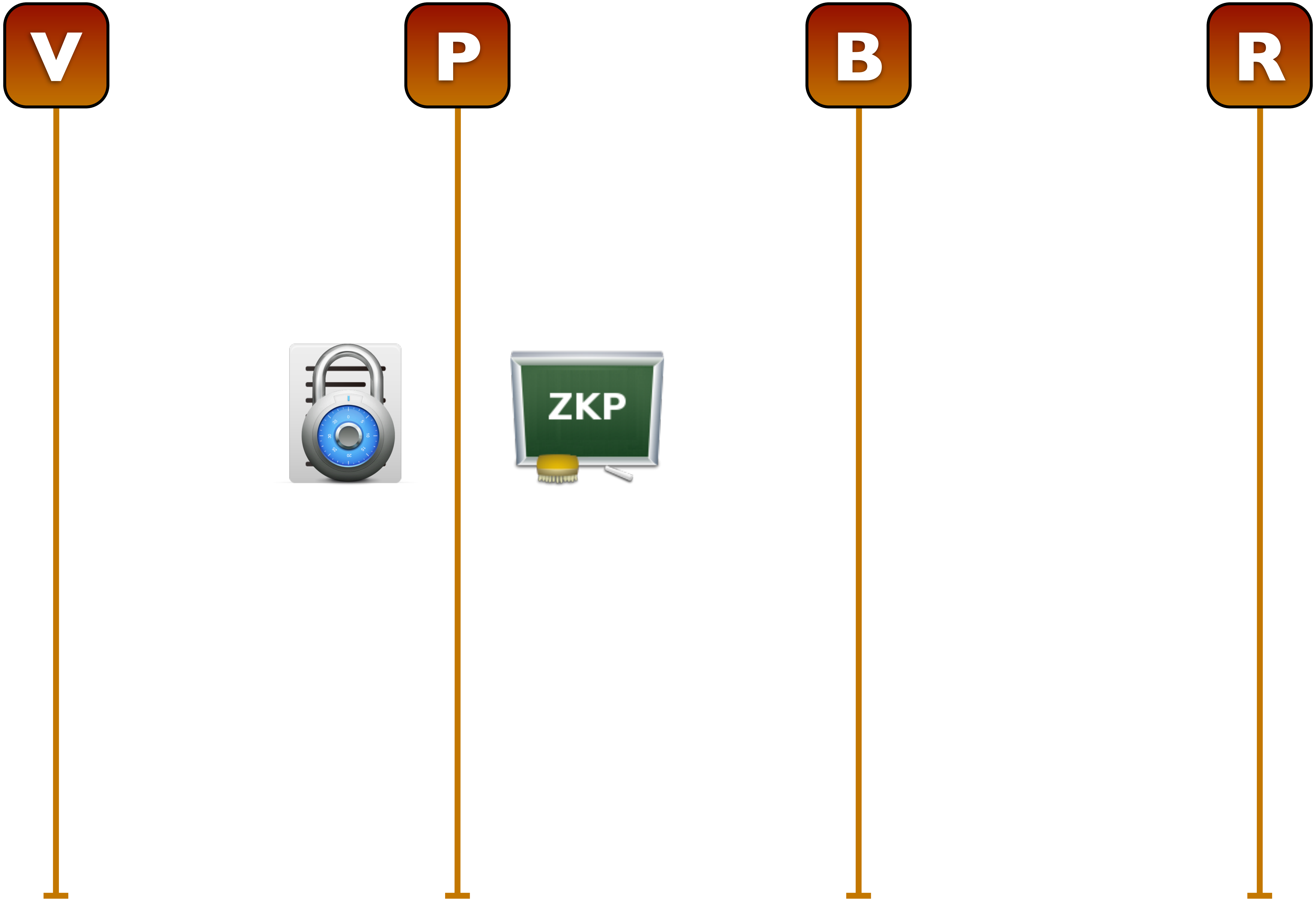
# Submission process



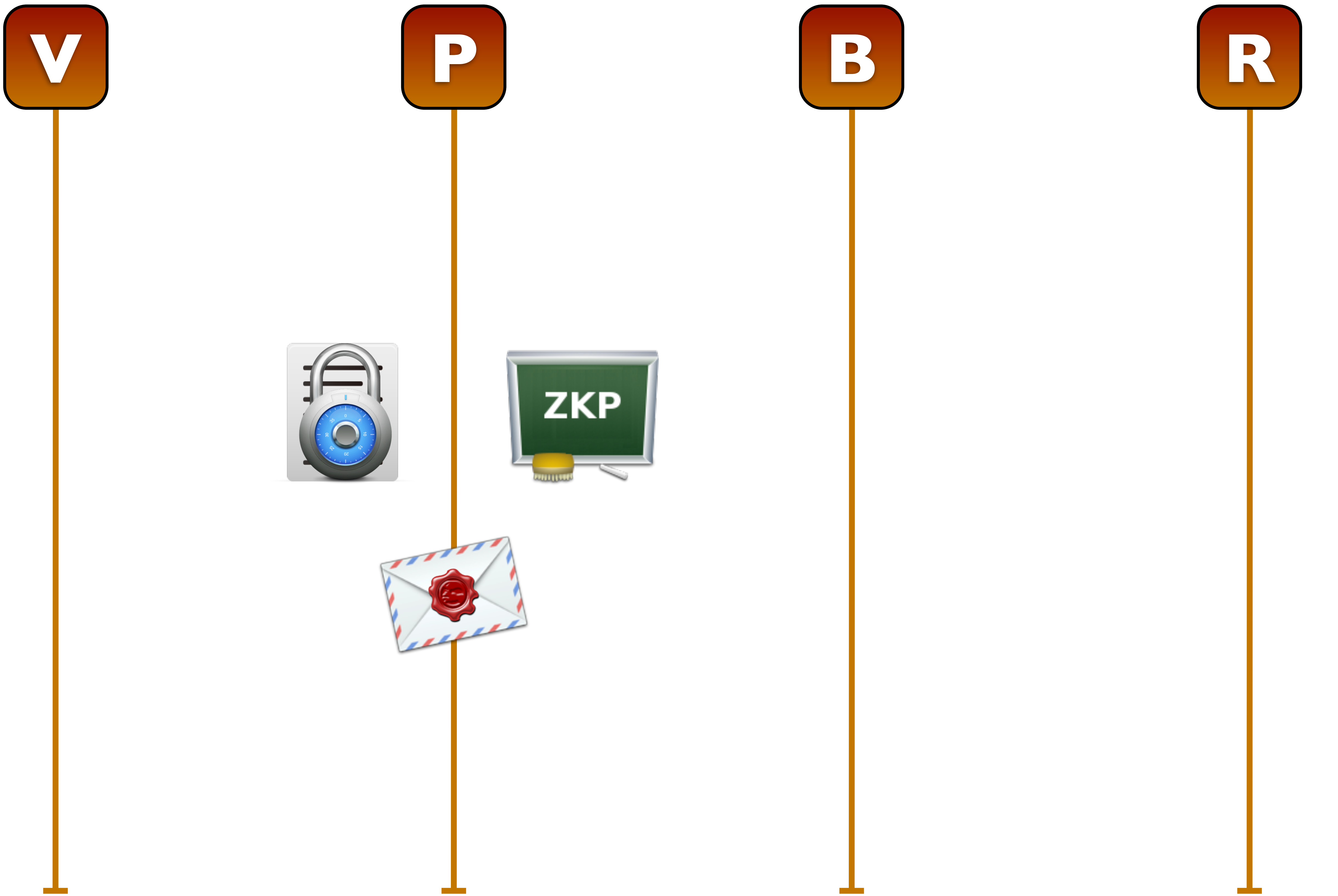
# Submission process



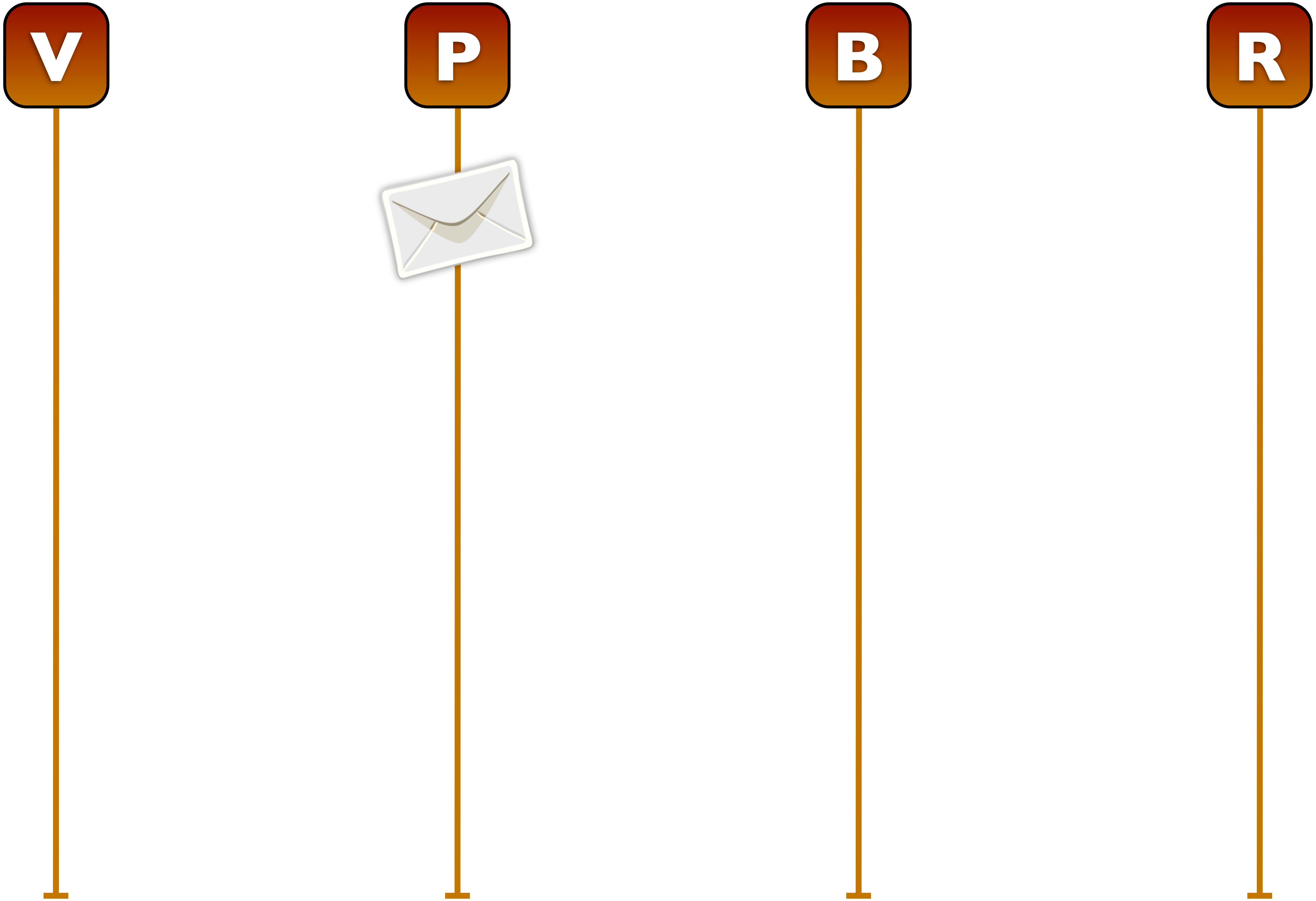
# Submission process



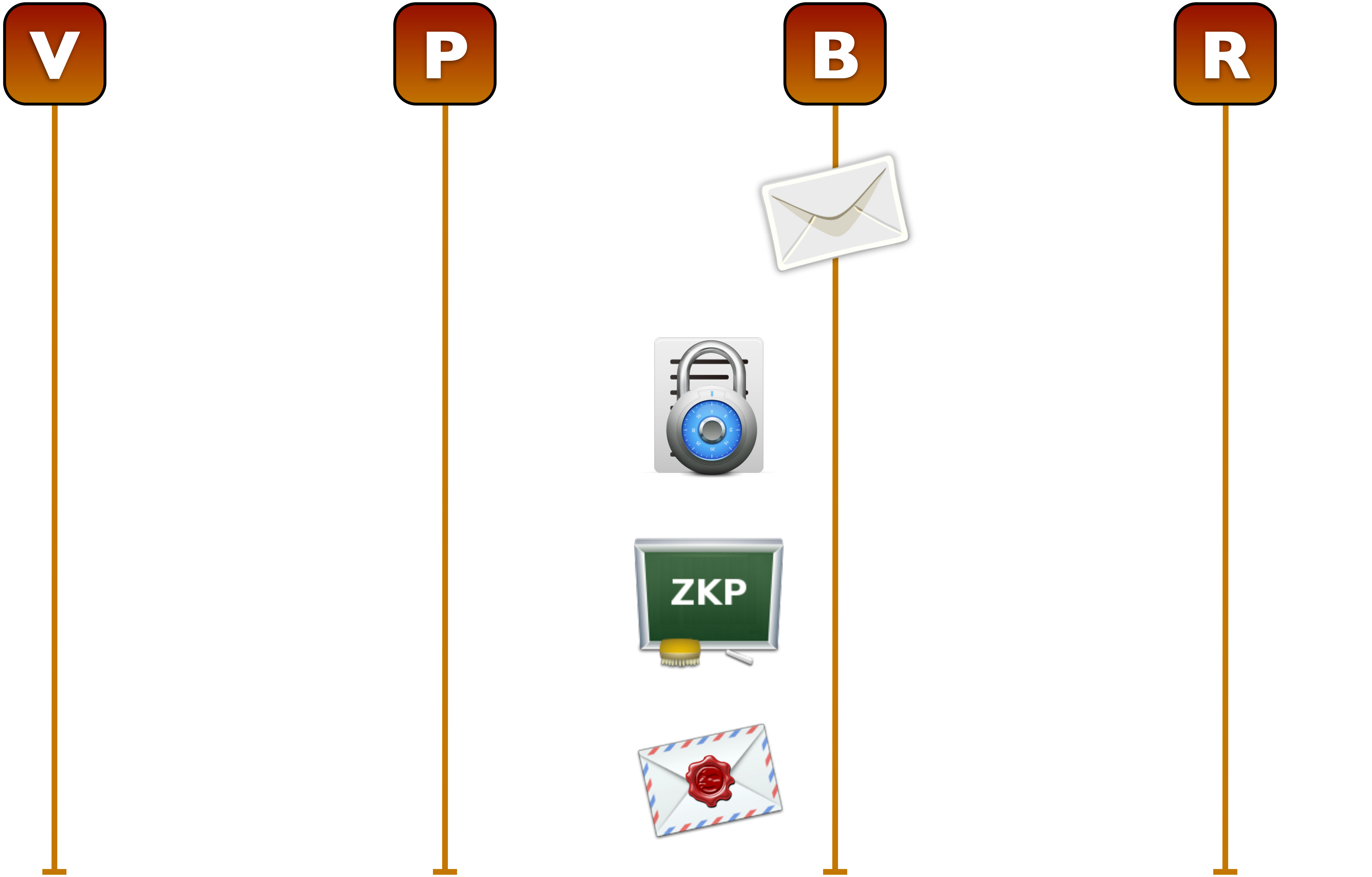
# Submission process



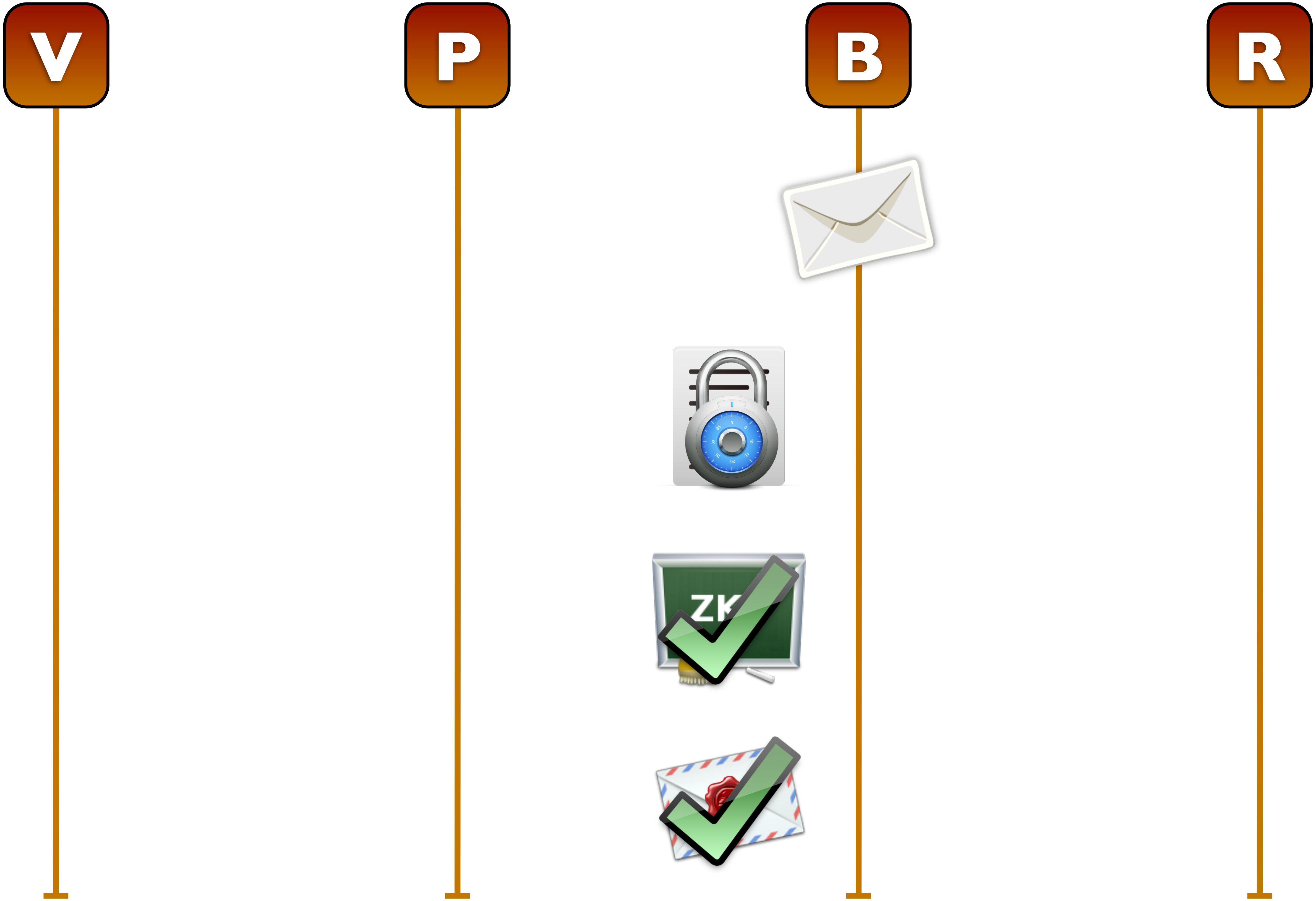
# Submission process



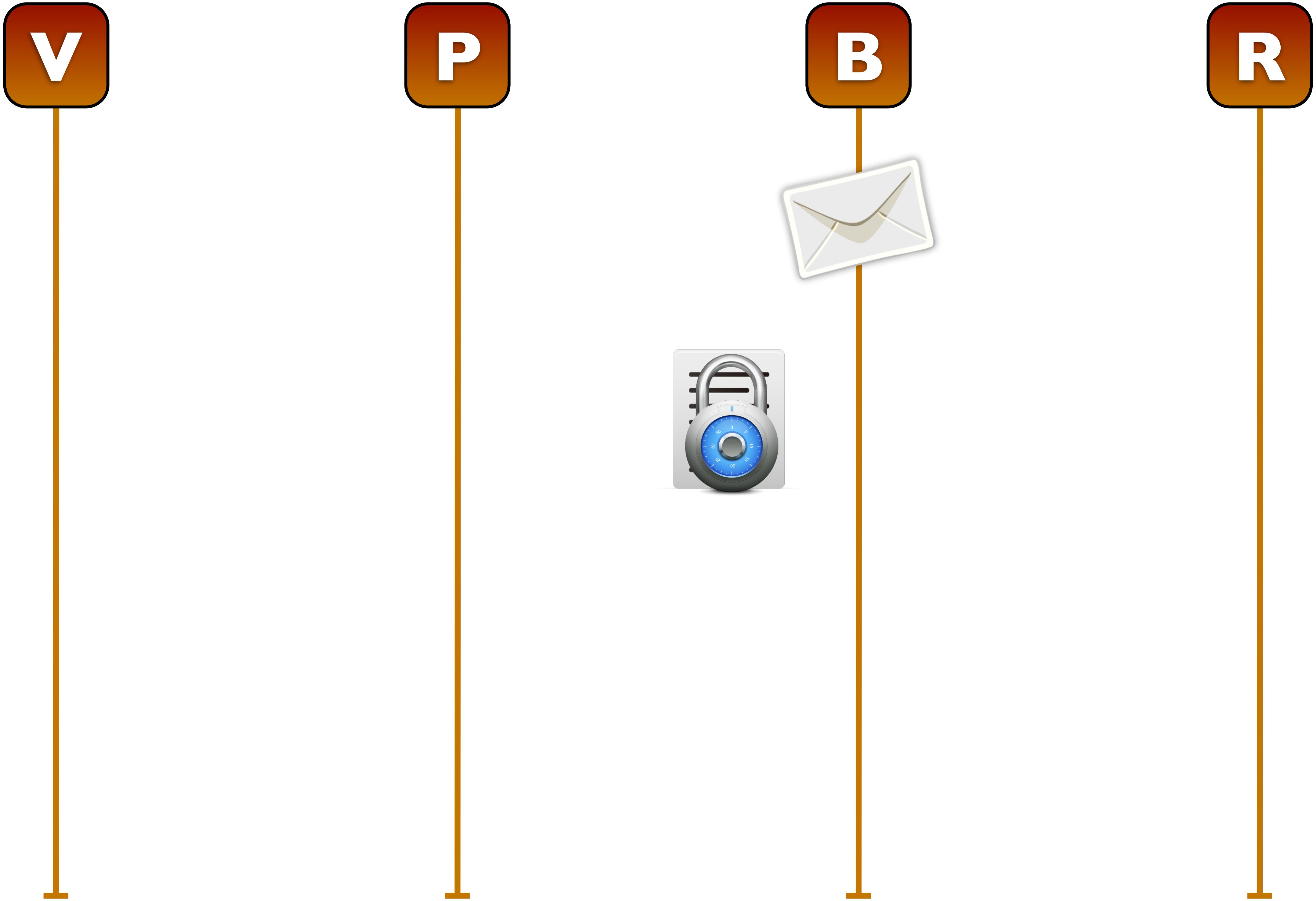
# Submission process



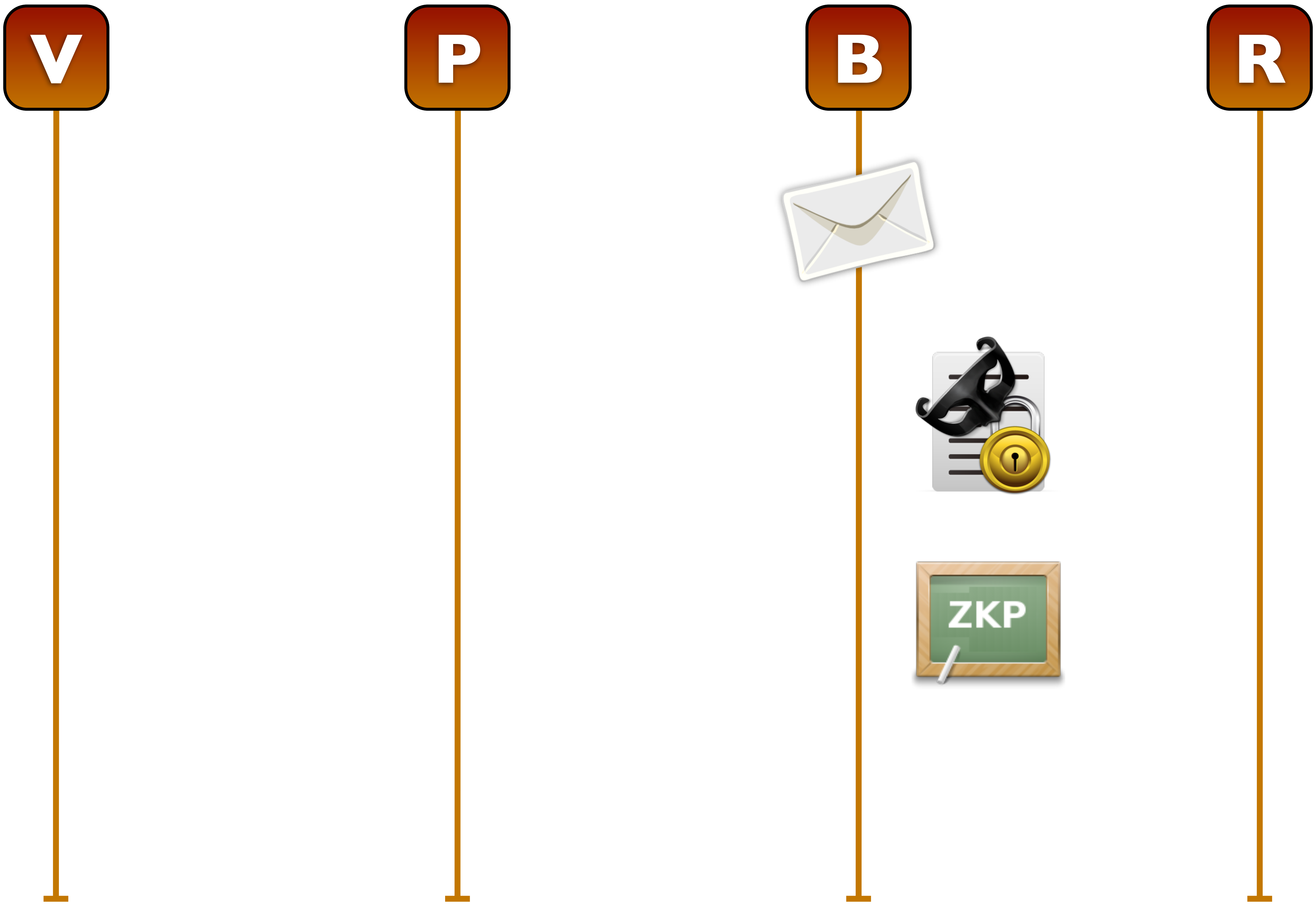
# Submission process



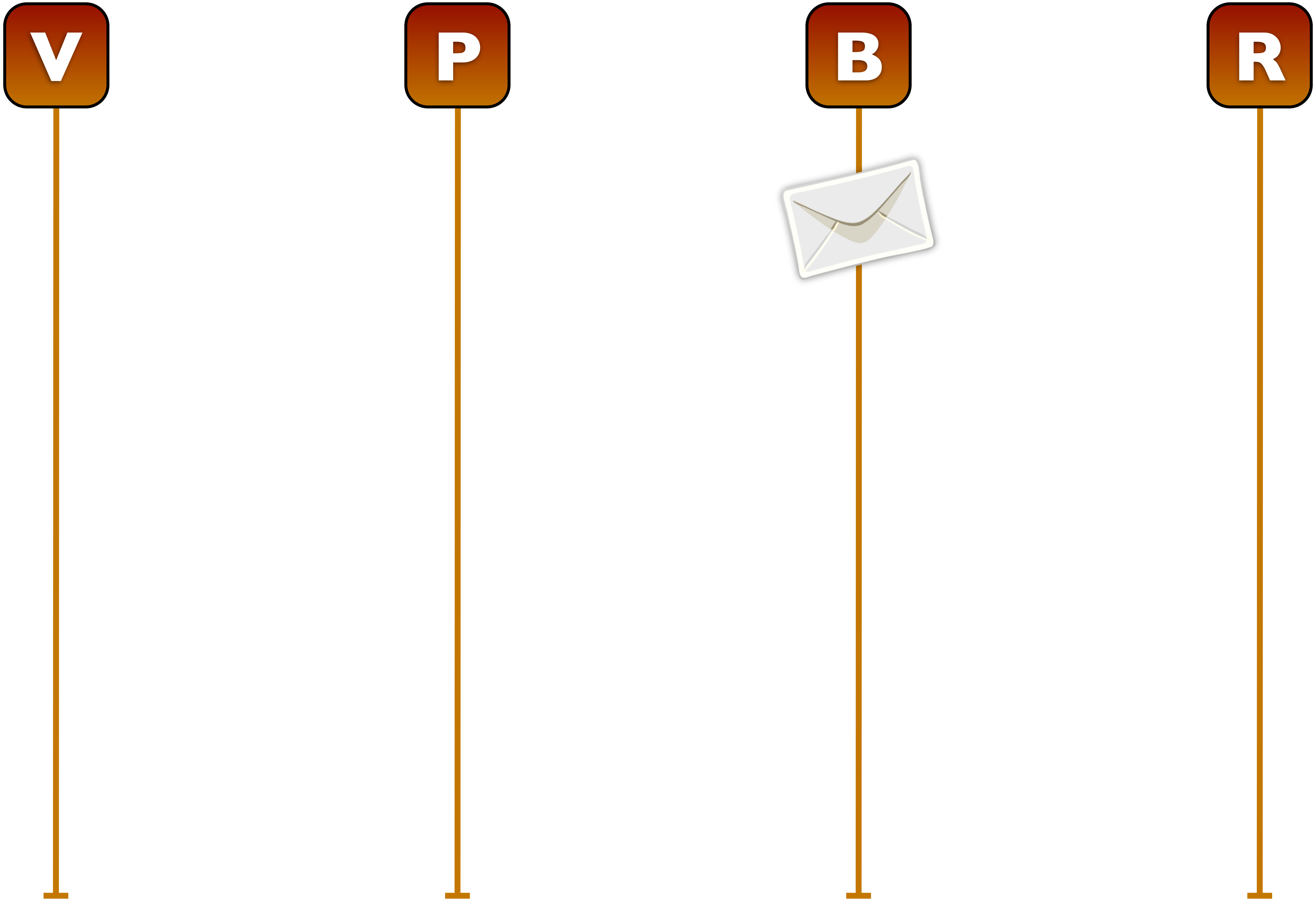
# Submission process



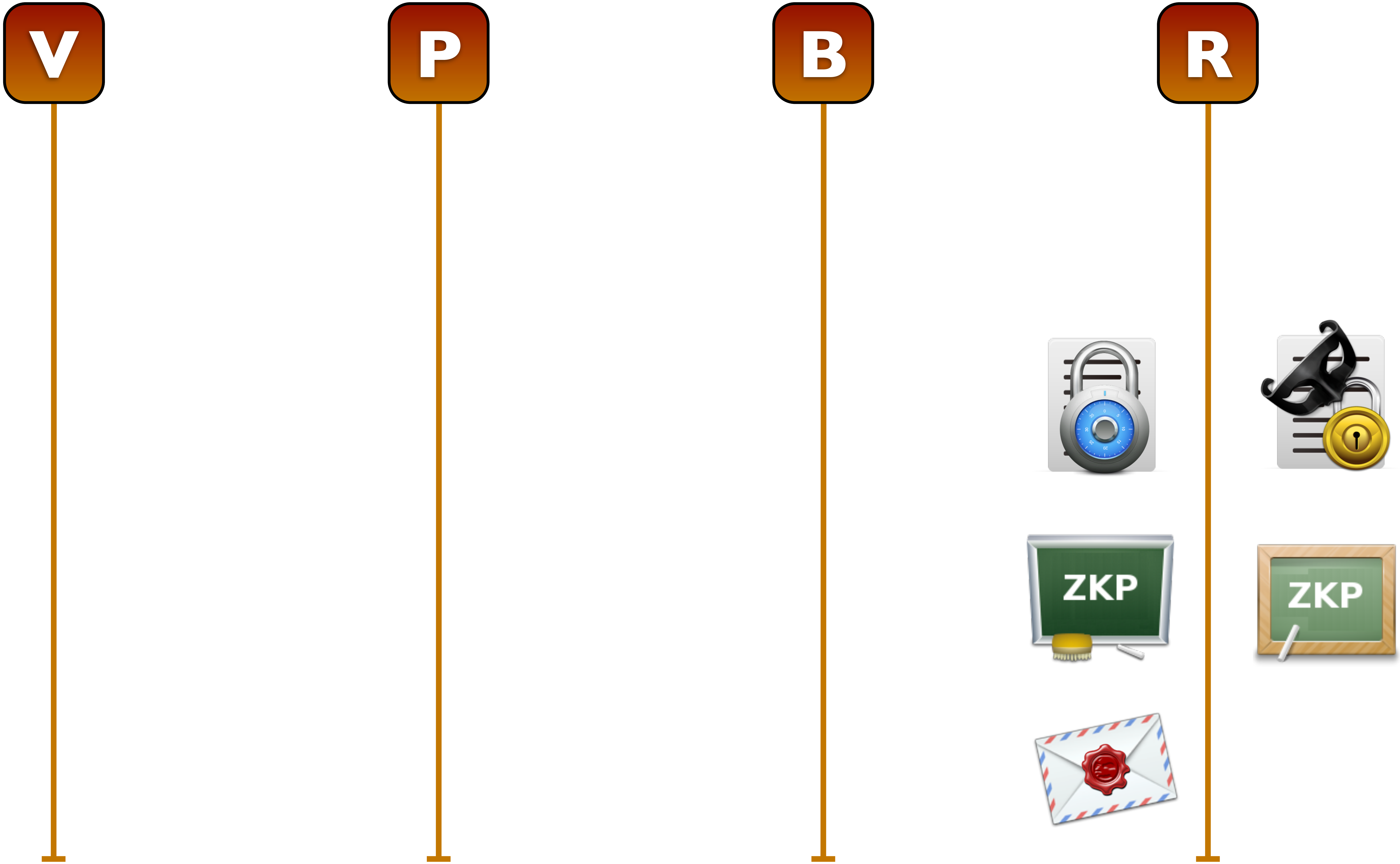
# Submission process



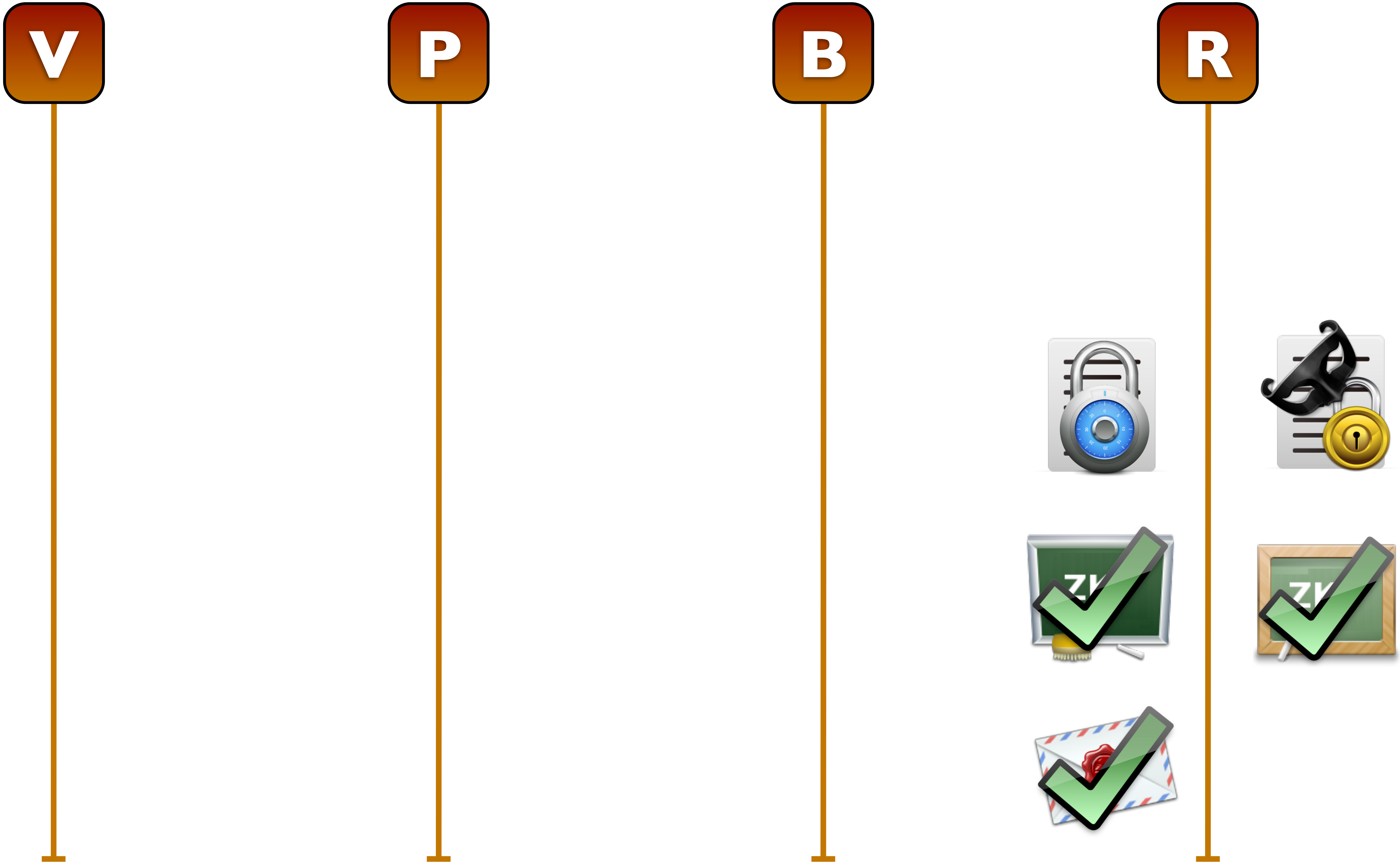
# Submission process



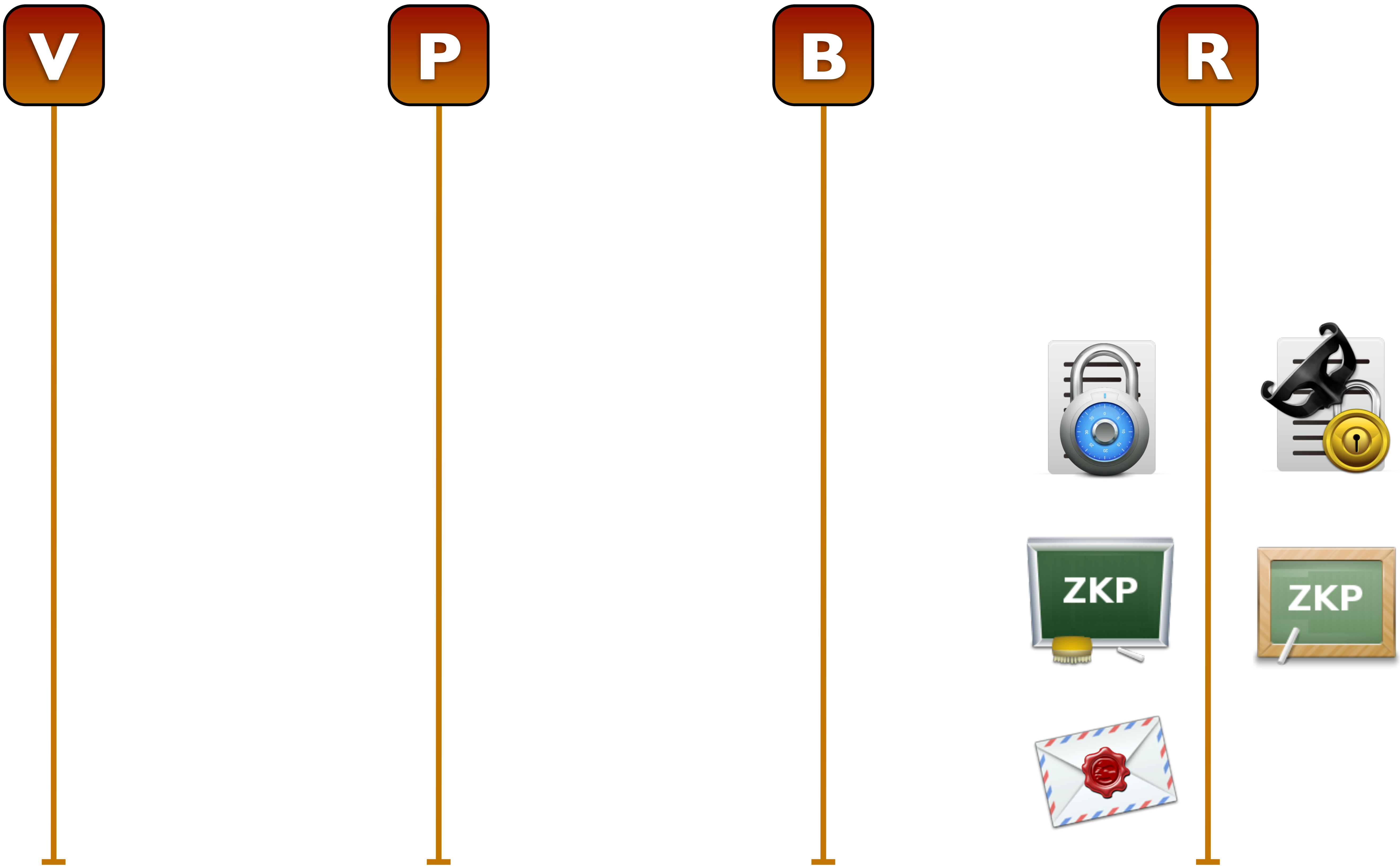
# Submission process



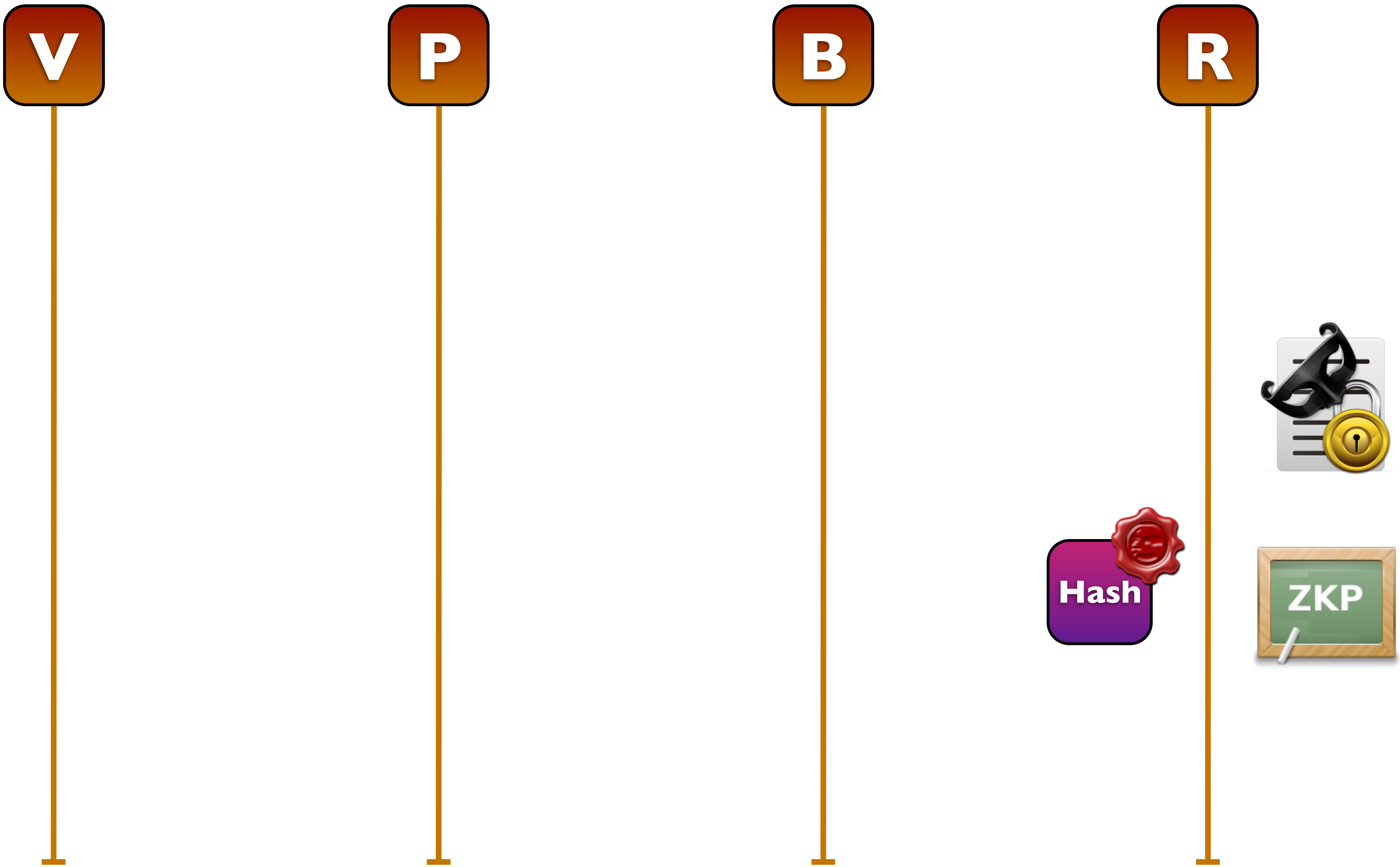
# Submission process



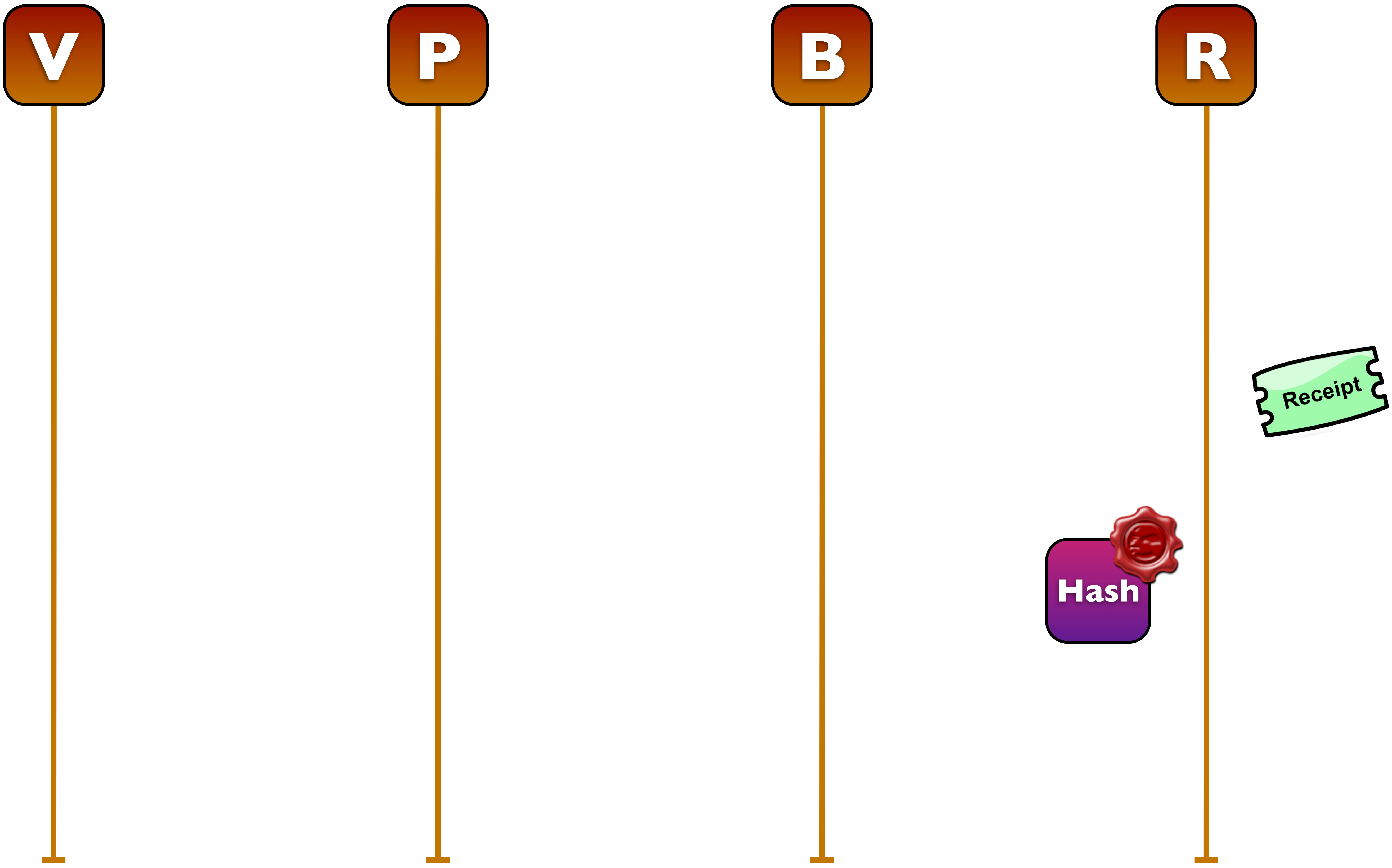
# Submission process



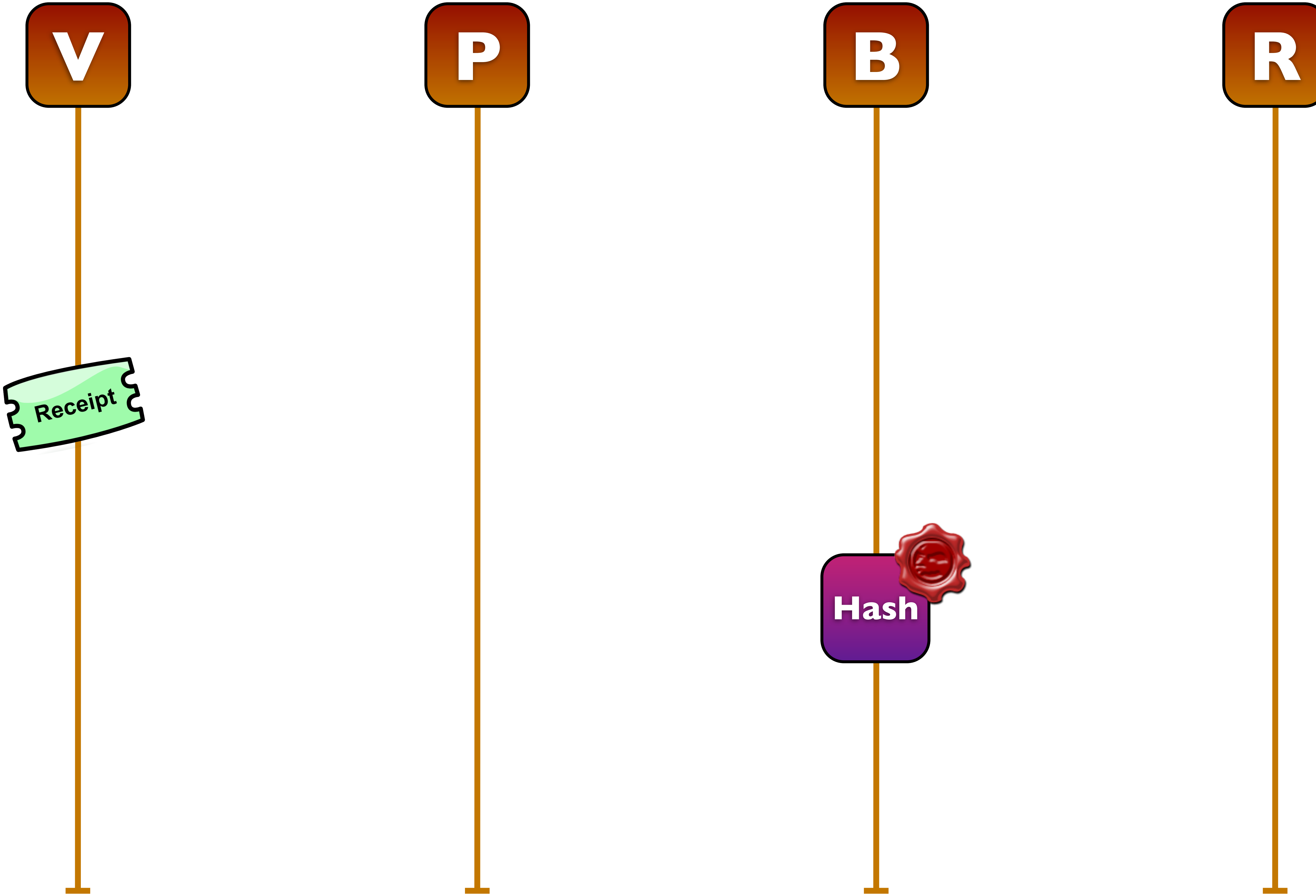
# Submission process



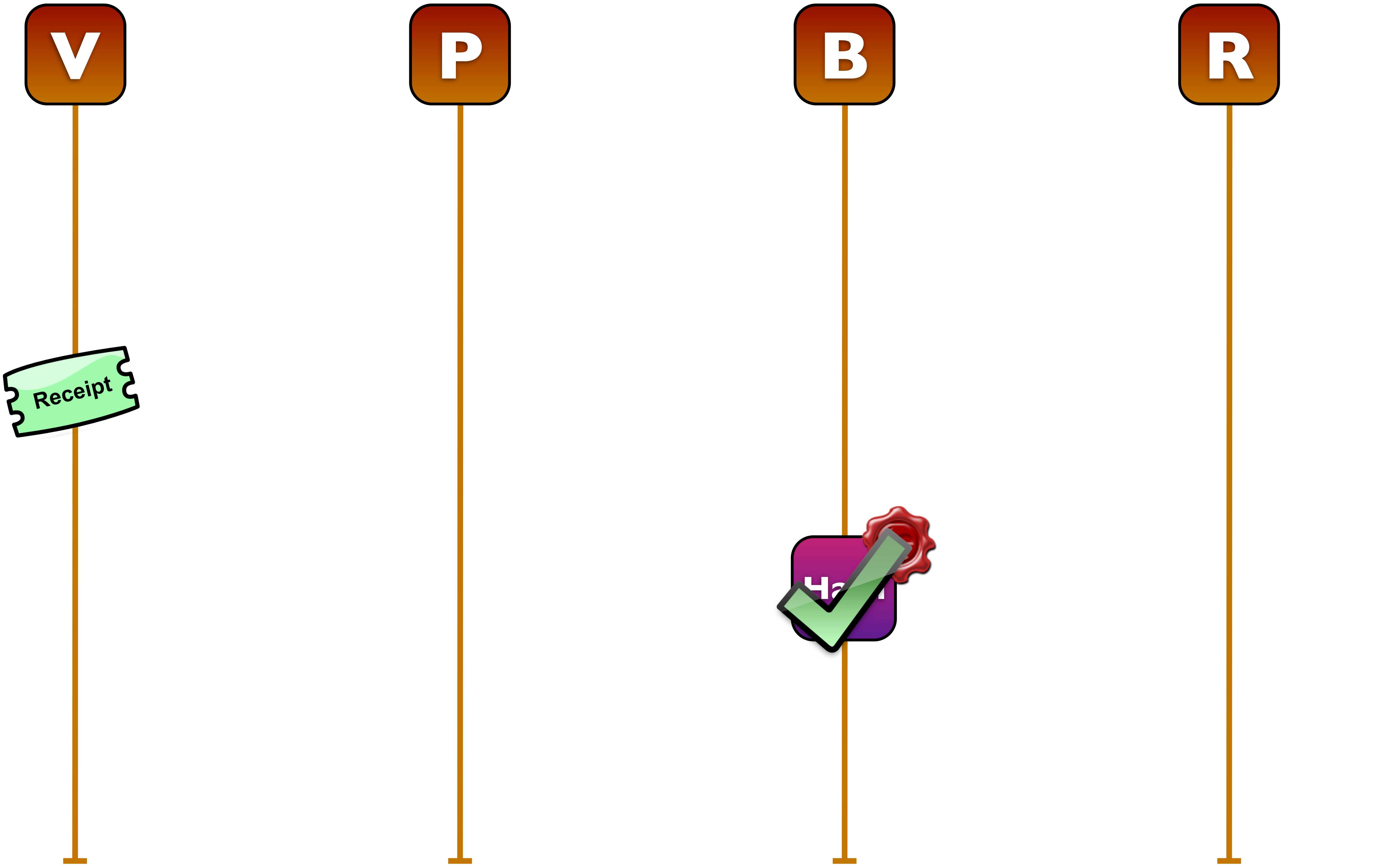
# Submission process



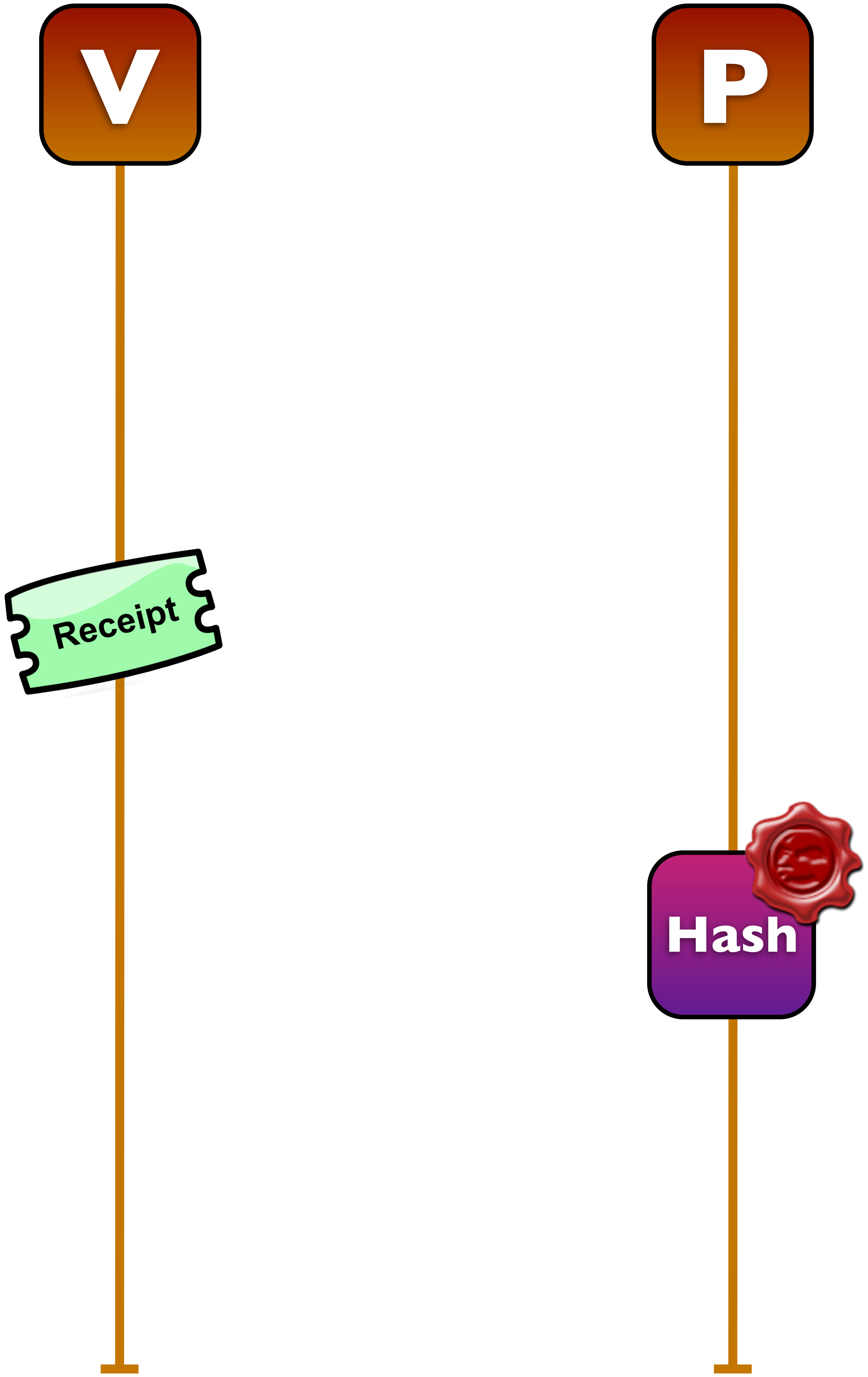
# Submission process



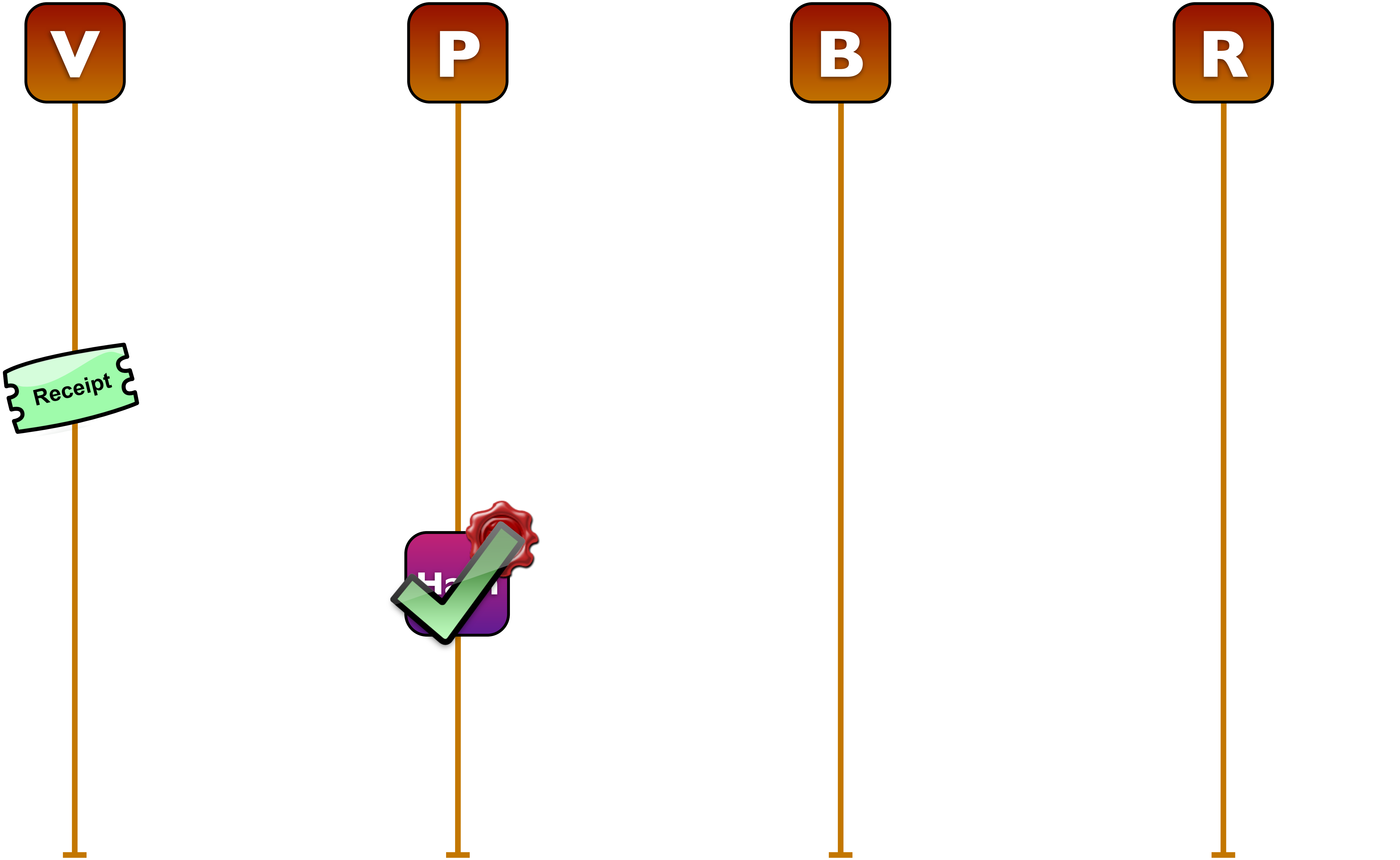
# Submission process



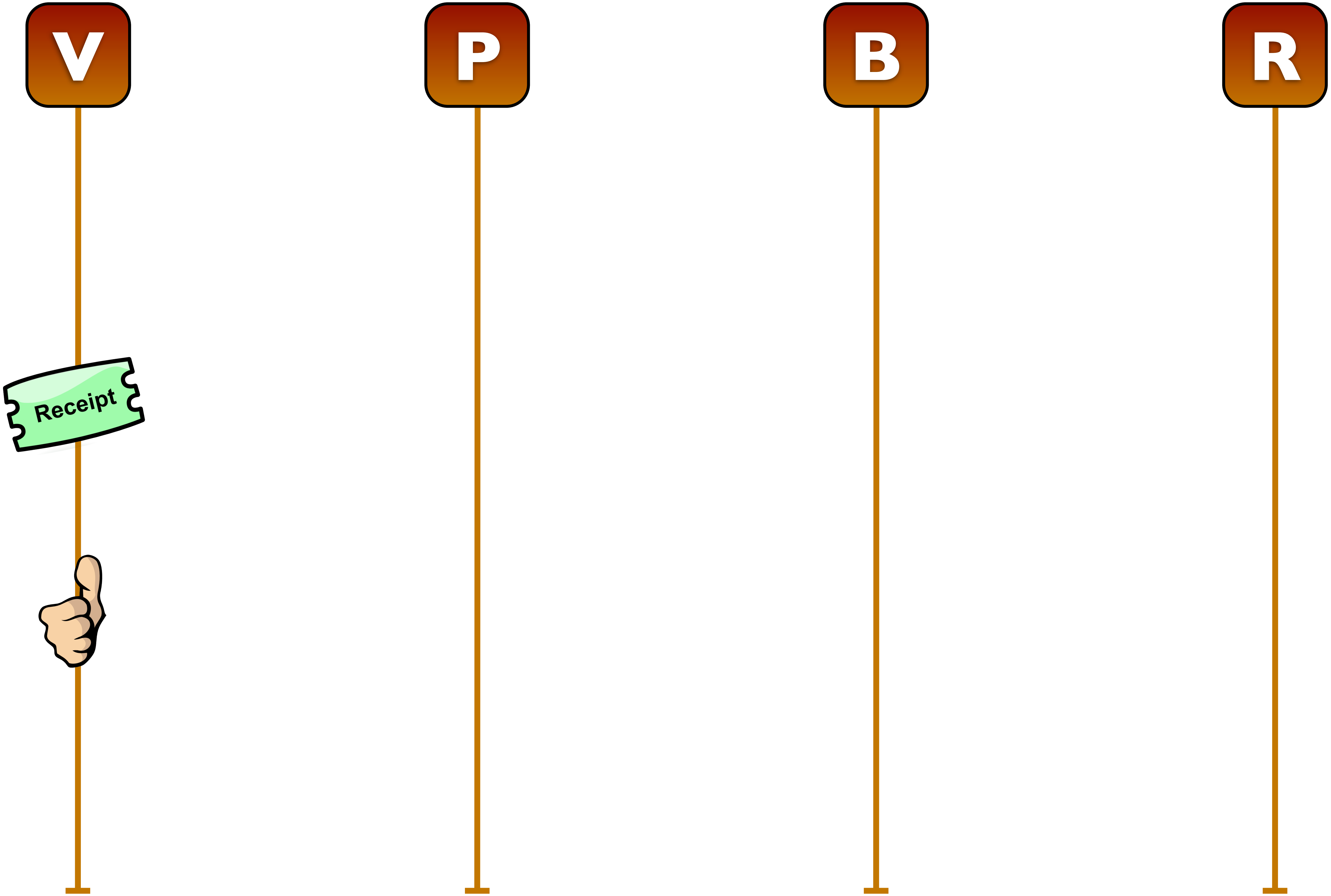
# Submission process



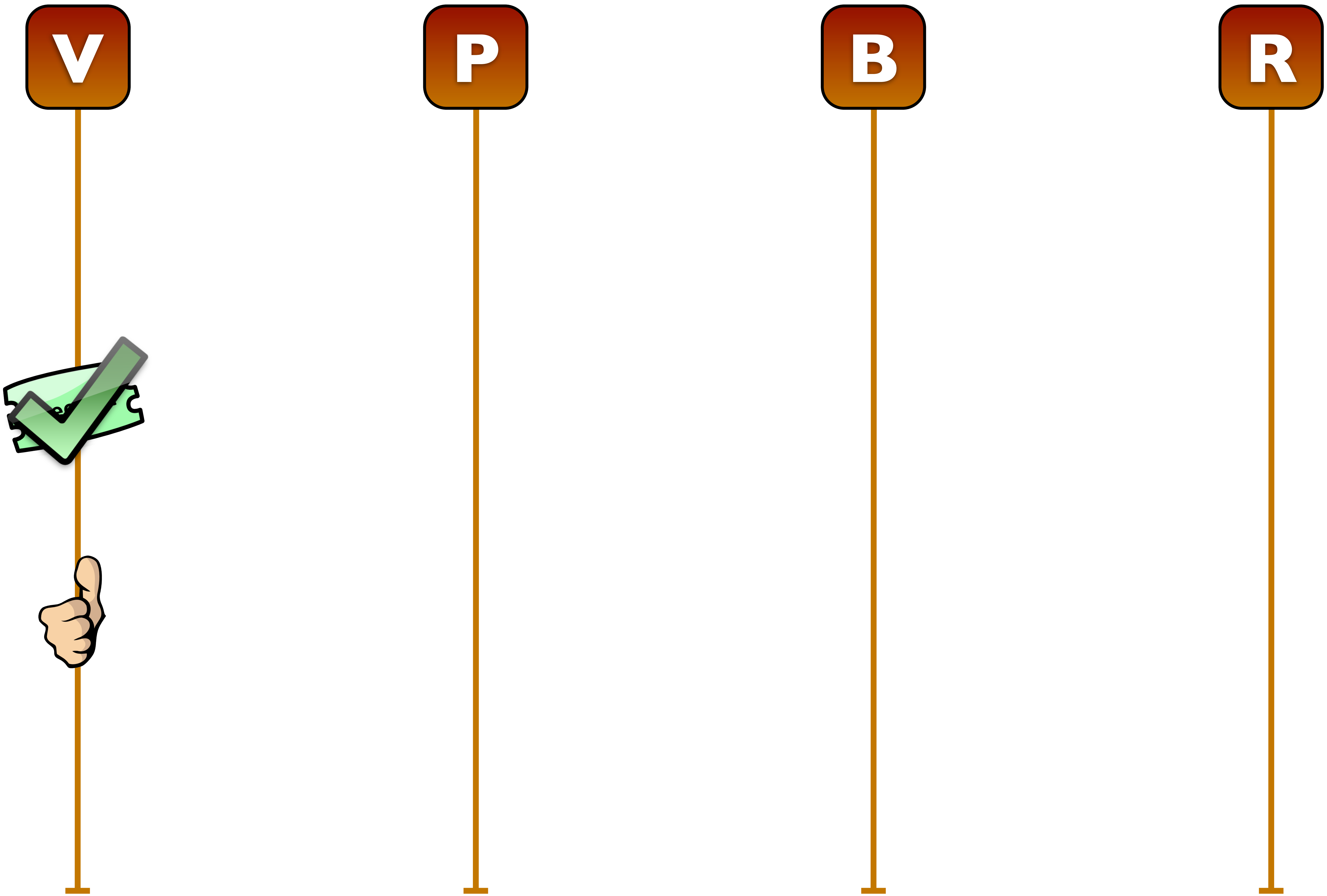
# Submission process



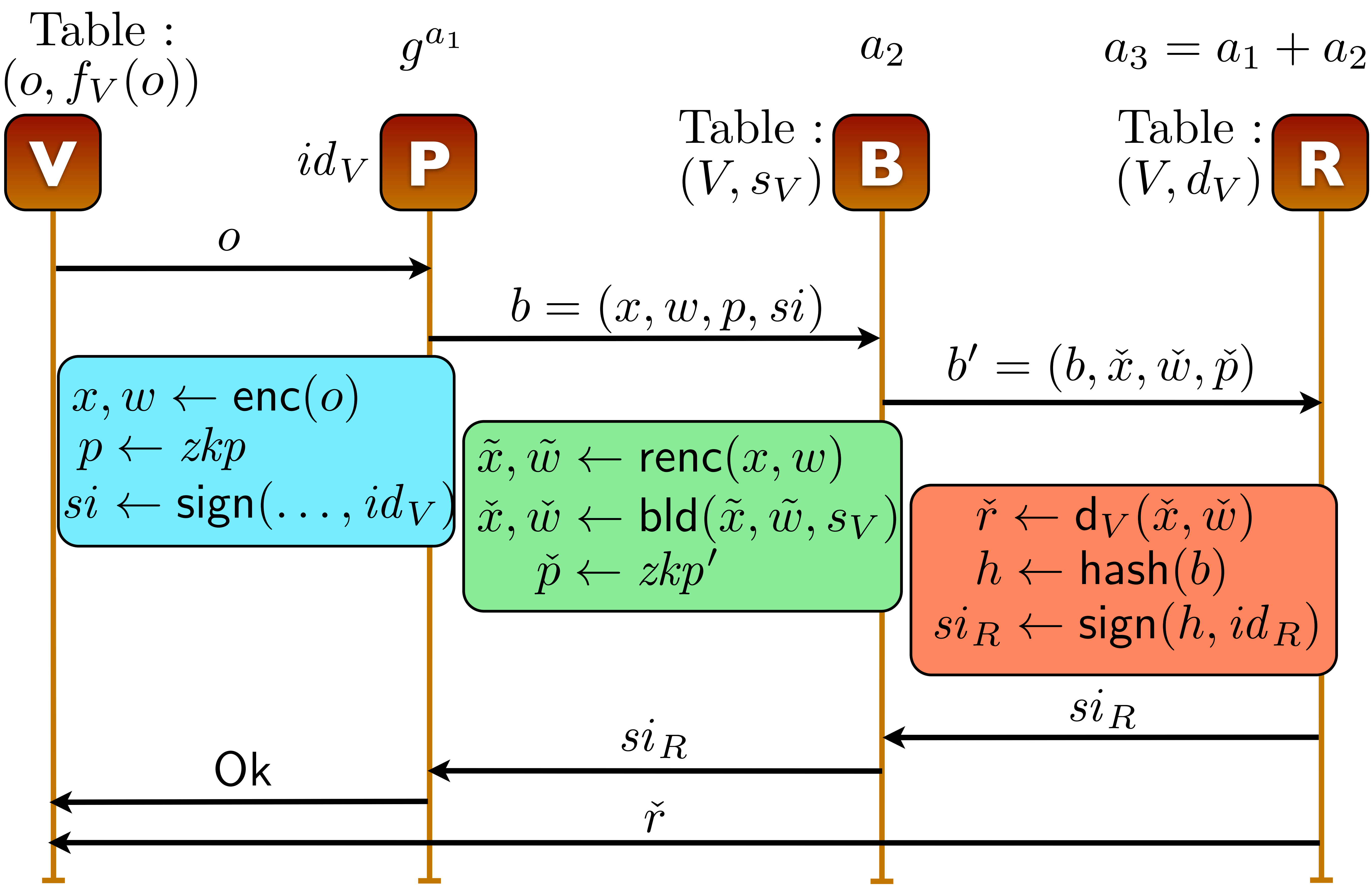
# Submission process



# Submission process



# Submission process

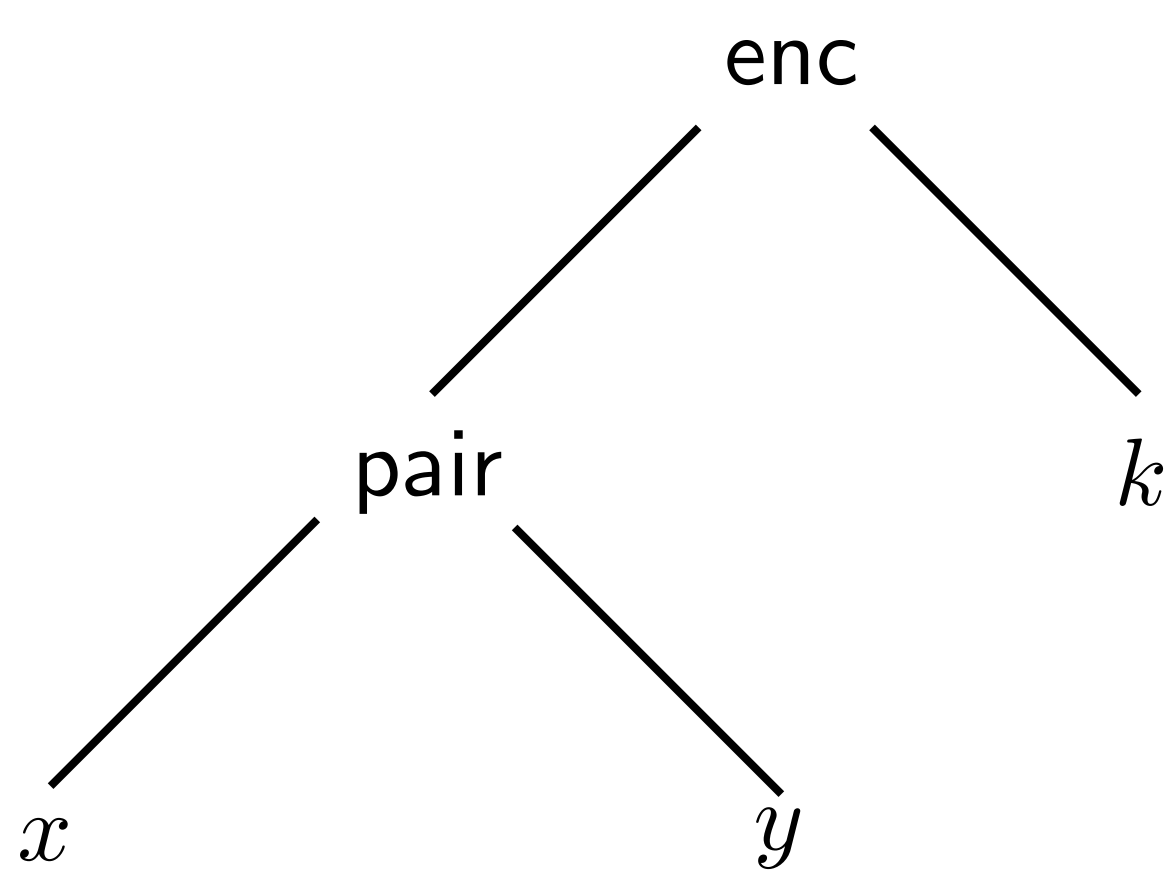


# Abstraction by terms

**Nonces :** $n, m, \ldots$ **Keys :** $k_1, \ldots, k_n, \ldots$

**Primitives :** $\mathsf{pair}(x, y), \mathsf{enc}(x, k), \mathsf{blind}(x, s), \ldots$

Message $\mathsf{enc}(\mathsf{pair}(x, y), k)$ is represented by :



# Equational theory

$$\mathsf{fst}(\mathsf{pair}(x, y)) = x \qquad \mathsf{snd}(\mathsf{pair}(x, y)) = y$$

$$\mathsf{dec}(\mathsf{penc}(x, r, \mathsf{pk}(k)), k) = x \qquad \mathsf{unblind}(\mathsf{blind}(x, s), s) = x$$

# Equational theory

$$\mathsf{fst}(\mathsf{pair}(x, y)) = x \qquad \mathsf{snd}(\mathsf{pair}(x, y)) = y$$

$$\mathsf{dec}(\mathsf{penc}(x, r, \mathsf{pk}(k)), k) = x \qquad \mathsf{unblind}(\mathsf{blind}(x, s), s) = x$$

$$\mathsf{dec}(\mathsf{blind}(\mathsf{penc}(x, r, \mathsf{pk}(k)), s), k) = \mathsf{blind}(x, s)$$

# Equational theory

$$\mathsf{fst}(\mathsf{pair}(x, y)) = x \qquad \mathsf{snd}(\mathsf{pair}(x, y)) = y$$

$$\mathsf{dec}(\mathsf{penc}(x, r, \mathsf{pk}(k)), k) = x \qquad \mathsf{unblind}(\mathsf{blind}(x, s), s) = x$$

$$\mathsf{dec}(\mathsf{blind}(\mathsf{penc}(x, r, \mathsf{pk}(k)), s), k) = \mathsf{blind}(x, s)$$

$$\mathsf{penc}(x_1, r_1, k_p) \circ \mathsf{penc}(x_2, r_2, k_p) = \mathsf{penc}(x_1 \diamond x_2, r_1 * r_2, k_p)$$

$$\mathsf{renc}(\mathsf{penc}(x, r, \mathsf{pk}(k_1)), k_2) = \mathsf{penc}(x, r, \mathsf{pk}(k_1 + k_2))$$

# Equational theory

$$\mathsf{fst}(\mathsf{pair}(x,y)) = x \qquad \mathsf{snd}(\mathsf{pair}(x,y)) = y$$

$$\mathsf{dec}(\mathsf{penc}(x,r,\mathsf{pk}(k)),k) = x \qquad \mathsf{unblind}(\mathsf{blind}(x,s),s) = x$$

$$\mathsf{dec}(\mathsf{blind}(\mathsf{penc}(x,r,\mathsf{pk}(k)),s),k) = \mathsf{blind}(x,s)$$

$$\mathsf{penc}(x_1,r_1,k_p) \circ \mathsf{penc}(x_2,r_2,k_p) = \mathsf{penc}(x_1 \diamond x_2, r_1 * r_2, k_p)$$

$$\mathsf{renc}(\mathsf{penc}(x,r,\mathsf{pk}(k_1)),k_2) = \mathsf{penc}(x,r,\mathsf{pk}(k_1+k_2))$$

$$\mathsf{checksign}(x,y,\mathsf{sign}(x,y)) = \mathsf{Ok}$$

$$\mathsf{checkpfk}_1(\mathsf{vk}(i),\mathsf{ball},\mathsf{pfk}_1(i,r,x,\mathsf{ball})) = \mathsf{Ok} \mid \mathsf{ball} = \mathsf{penc}(x,r,k_p)$$

$$\mathsf{checkpfk}_2(\mathsf{vk}(i),\mathsf{ball},\mathsf{pfk}_2(i,r,x,\mathsf{ball})) = \mathsf{Ok} \mid \begin{array}{c}\mathsf{ball} = \mathsf{renc}(x,r)\\ \mathsf{ball} = \mathsf{blind}(x,r)\end{array}$$

# Applied Pi-Calculus

| | |
|---|---|
| $P, Q, R ::=$ | (plain) processes |
| $0$ | null process |
| $P \mid Q$ | parallel composition |
| $!P$ | replication |
| $\nu\, n.P$ | name restriction |
| $\text{if } \phi \text{ then } P \text{ else } Q$ | conditional |
| $u(x).P$ | message input |
| $\overline{u}\langle M \rangle.P$ | message output |

Introduced by
Abadi and Fournet

| | |
|---|---|
| $A, B, C ::=$ | extended processes |
| $P$ | plain process |
| $A \mid B$ | parallel composition |
| $\nu\, n.A$ | name restriction |
| $\nu\, x.A$ | variable restriction |
| $\{^{M}/_{x}\}$ | active substitution |

# Modeling of players

**Example :** Modeling of the voter

$$
\begin{array}{l}
V(c_{auth}, c_{out}, c_{RV}, g_1, id, idp_R, x_{vote}) = \nu\, t\ . \\
\quad \text{let } e = \mathsf{penc}(x_{vote}, t, g_1) \text{ in} \\
\quad \text{let } p = \mathsf{pfk}_1(id, t, x_{vote}, e) \text{ in} \\
\quad \text{let } si = \mathsf{sign}((e, p), id) \text{ in} \\
\quad \overline{c_{out}}\langle (e, p, si) \rangle\ . \\
\quad \overline{c_{auth}}\langle (e, p, si) \rangle\ . \\
\quad c_{RV}(x)\ .\ c_{auth}(y)\ . \\
\quad \overline{c_{out}}\langle x \rangle\ .\ \overline{c_{out}}\langle y \rangle\ . \\
\quad \text{let } hv = \mathsf{hash}((\mathsf{vk}(id), e, p, si)) \text{ in} \\
\quad \text{if } \phi_{\mathsf{v}}(idp_R, id, h, x, x_{vote}, y) \text{ then } \overline{c_{auth}}\langle \mathsf{Ok} \rangle
\end{array}
$$

# Vote-Privacy

**Definition :** *Vote-Privacy* (Delaune, Kremer & Ryan)

A voting protocol ensures vote-privacy if :

$$S[V_A\{^{\mathbf{v_1}}/_{v}\} \mid V_B\{^{\mathbf{v_2}}/_{v}\}] \approx_l S[V_A\{^{\mathbf{v_2}}/_{v}\} \mid V_B\{^{\mathbf{v_1}}/_{v}\}]$$

# Vote-Privacy

**Definition :** *Vote-Privacy* (Delaune, Kremer & Ryan)

A voting protocol ensures vote-privacy if :

$$S[V_A\{^{\mathbf{v_1}}/_v\} \mid V_B\{^{\mathbf{v_2}}/_v\}] \approx_l S[V_A\{^{\mathbf{v_2}}/_v\} \mid V_B\{^{\mathbf{v_1}}/_v\}]$$

**How can we prove this ?**

- Using ProVerif ? (or another automatic tool)

**No** ⇒ The equational theory is too complex to be handled by ProVerif. (or any existing tool.)

- We have to do this by hand.

# Results

Assuming that all **infrastructure players** are **honest**...

**Theorem**

Vote-privacy with only 2 honest voters :

$$S[V_A\{^{\mathbf{v_1}}/_{x_v}\} \mid V_B\{^{\mathbf{v_2}}/_{x_v}\}] \approx_l S[V_A\{^{\mathbf{v_2}}/_{x_v}\} \mid V_B\{^{\mathbf{v_1}}/_{x_v}\}]$$

**Theorem**

Vote-privacy with only 2 honest voters and **without auditor** :

$$S'[V_A\{^{\mathbf{v_1}}/_{x_v}\} \mid V_B\{^{\mathbf{v_2}}/_{x_v}\}] \approx_l S'[V_A\{^{\mathbf{v_2}}/_{x_v}\} \mid V_B\{^{\mathbf{v_1}}/_{x_v}\}]$$

# Sketch of proof

**Two steps proof :**

- **Step 1 - Finding a bisimulation**

  1 - Representing  all possible successors of the two processes.

  2 - Giving a relation R and proving that it is a bisimulation.

# Sketch of proof

**Two steps proof :**

- **Step 1 - Finding a bisimulation**

  1 - Representing  all possible successors of the two processes.

  2 - Giving a relation R and proving that it is a bisimulation.

- **Step 2 - Static equivalence property**

  Proving that two (big) final frames are in static equivalence.

# Sketch of proof

- **Step 2.a -** Only a **limited (but infinite)** number of static equivalences needs to be considered.

# Sketch of proof

- **Step 2.a -** Only a **limited (but infinite)** number of static equivalences needs to be considered.

**Lemma (simplified)**

$\forall M_i$ (i=3,n) deducible from messages :

$$\{^{ballot_1^{v1}}/_{x_1}, {}^{ballot_2^{v2}}/_{x_2}, {}^{\mathsf{d}_i(\mathsf{dec}(\mathsf{blind}(\mathsf{renc}(M_i,a_2),s_i),a_3))}/_{y_i}, {}^{\mathsf{sign}(\mathsf{hash}(vk_i,M_i),id_R)}/_{z_i}, {}^{\mathsf{dec}(\Pi_1(M_i))}/_{res_i}, i=3,n\}$$

$$\approx_s \{^{ballot_1^{v2}}/_{x_1}, {}^{ballot_2^{v1}}/_{x_2}, {}^{\mathsf{d}_i(\mathsf{dec}(\mathsf{blind}(\mathsf{renc}(M_i,a_2),s_i),a_3))}/_{y_i}, {}^{\mathsf{sign}(\mathsf{hash}(vk_i,M_i),id_R)}/_{z_i}, {}^{\mathsf{dec}(\Pi_1(M_i))}/_{res_i}, i=3,n\}$$

# Sketch of proof

- **Step 2.a -** Only a **limited (but infinite)** number of static equivalences needs to be considered.

**Lemma (simplified)**

$\forall M_i$ (i=3,n) deducible from messages :

$$\{^{ballot_1^{v1}}/_{x_1}, ^{ballot_2^{v2}}/_{x_2}, ^{\mathsf{d}_i(\mathsf{dec}(\mathsf{blind}(\mathsf{renc}(M_i,a_2),s_i),a_3))}/_{y_i}, ^{\mathsf{sign}(\mathsf{hash}(vk_i,M_i),id_R)}/_{z_i}, ^{\mathsf{dec}(\Pi_1(M_i))}/_{res_i}, i = 3,n\}$$

$$\approx_s \{^{ballot_1^{v2}}/_{x_1}, ^{ballot_2^{v1}}/_{x_2}, ^{\mathsf{d}_i(\mathsf{dec}(\mathsf{blind}(\mathsf{renc}(M_i,a_2),s_i),a_3))}/_{y_i}, ^{\mathsf{sign}(\mathsf{hash}(vk_i,M_i),id_R)}/_{z_i}, ^{\mathsf{dec}(\Pi_1(M_i))}/_{res_i}, i = 3,n\}$$

- **Step 2.b -** Using (and proving) **independence** lemmas :
  - $\Phi_1 \approx_s \Phi_2 \Rightarrow \Phi_1 \cup \{^{\mathsf{sign}(M,s)}/_t\} \approx_s \Phi_2 \cup \{^{\mathsf{sign}(M,s)}/_t\}$
  - $\Phi_1 \approx_s \Phi_2 \Rightarrow \Phi_1 \cup \{^{\mathsf{dec}(M,k)}/_t\} \approx_s \Phi_2 \cup \{^{\mathsf{dec}(M,k)}/_t\}$

# ProVerif & ProSwapper

**Use** of **ProVerif** in order to test further cases of corruption.

Only on a **simplified** equational theory (no AC-symbols).

→ We may miss some attacks but it is still interesting.

# Results

| Corr. Admin. Players \ Corr. Voters | 0 | 2 | 4 |
|---|---|---|---|
| None | | ✓ | |
| Ballot Box (B) | | ? | |
| Receipt Generator (R) | | ✓ | |
| Decrypt. Service (D)* | | ✓ | |
| Auditor (A) | | ✓ | |
| R+D* | | ✓ | |
| R+A | | ✓ | |
| B+R, B+R+A, B+D, B+D+A | | ✗ | |

# Moral of the story

We can have **some confidence** in the Norwegian protocol.

# Moral of the story

We can have **some confidence** in the Norwegian protocol.

**But** the study reveals some **crucial assumptions** :

- There should be **no virus** on the computer.

# Moral of the story

We can have **some confidence** in the Norwegian protocol.

**But** the study reveals some **crucial assumptions** :

- There should be **no virus** on the computer.
- **« Secure channels »** between infrastructure players :
  - Ballot box and Receipt generator,
  - Ballot box and Decryption device.

# Moral of the story

We can have **some confidence** in the Norwegian protocol.

**But** the study reveals some **crucial assumptions** :

- There should be **no virus** on the computer.
- **« Secure channels »** between infrastructure players :
  - Ballot box and Receipt generator,
  - Ballot box and Decryption device.
- How **initial secrets** are distributed ? By who ?
  - Secret keys,
  - Tables for Ballot Box, Receipt generator and voters.

# Conclusion

- A result on vote privacy of an implemented and deployed protocol.
- Some interesting results on corruption scenarios.
- Useful properties for next studies of protocols or the development of an automatic tool.

# Conclusion & Future work

- A result on vote privacy of an implemented and deployed protocol.
- Some interesting results on corruption scenarios.
- Useful properties for next studies of protocols or the development of an automatic tool.

- An analysis, by hand, of the case where the ballot box is corrupted.
- Study of properties like receipt-freeness, coercion-resistance, verifiability, ...
- Trying to develop an automatic tool capable of dealing with quite complicated equational theories to avoid such (exhausting) proofs.

# Thank you for your attention