How to Do Things with Cryptographic Protocols

Joshua D. Guttman

The MITRE Corporation

Thanks to the MITRE-Sponsored Research program

Asian Computer Science Conference, 2007

Protocols coordinate distributed systems

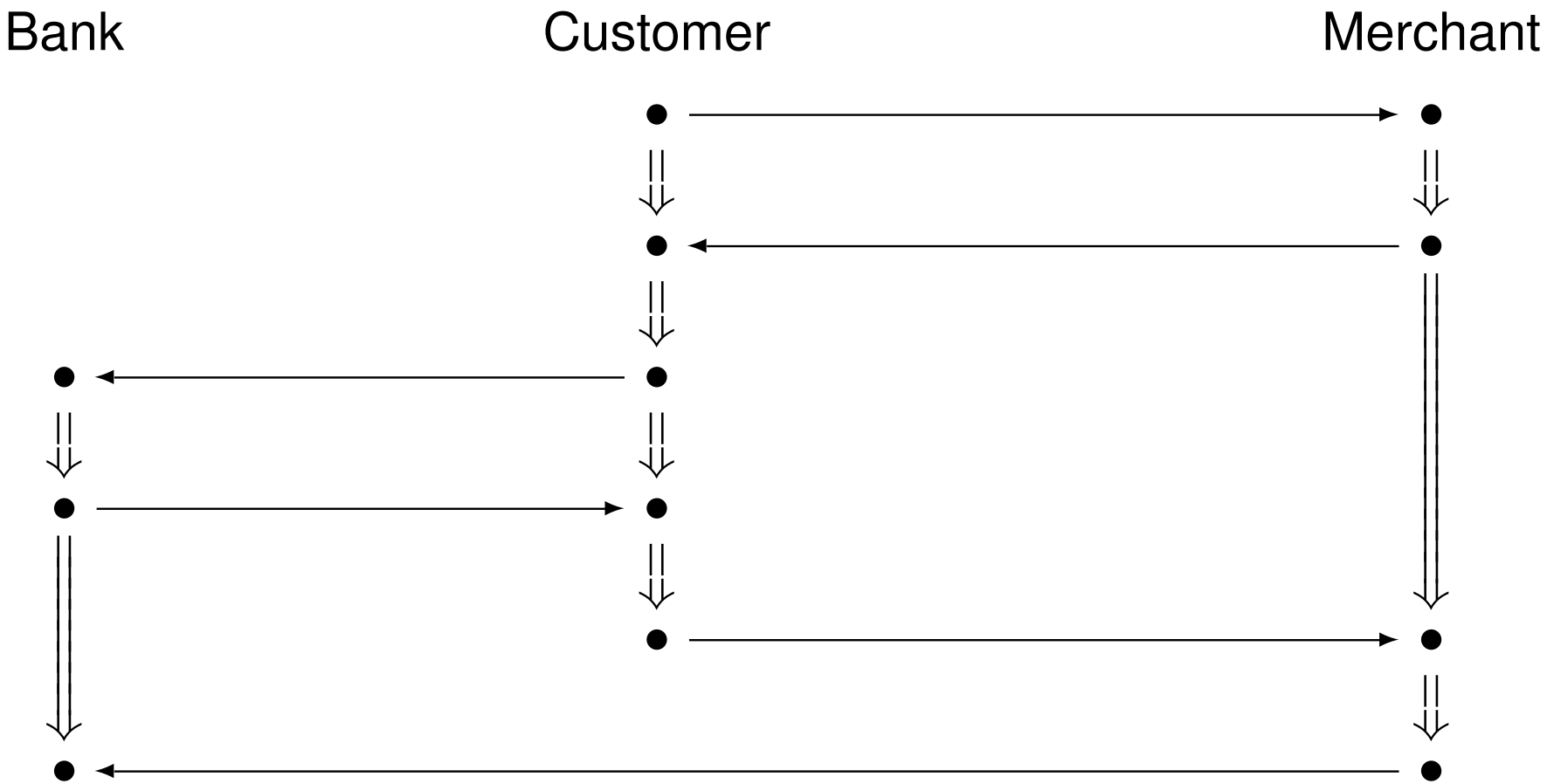
Protocols allow principals

- To agree on values
  - While preserving their secrecy
- To accept, select, or refuse commitments
  Often dependent on
  - Commitments received
  - Current local state
- To coordinate state changes
  between themselves

Despite presence of malicious adversaries

Electronic purchase

Using a money order (EPMO protocol)



EPMO Goals

- Agree on values
- Preserve confidentiality
- Allow decision-making
- Cause state change

EPMO Goals

- Agree on values:
  - C, M, B agree on each other's identities and price
  - C, M agree on goods; C, B agree on account number
- Preserve confidentiality
- Allow decision-making
- Cause state change

EPMO Goals

- Agree on values:
  - C, M, B agree on each other's identities and price
  - C, M agree on goods; C, B agree on account number
- Preserve confidentiality: Protect
  - C's account number from M, outsiders
  - goods from B, outsiders
  - price from outsiders
  - M's identity from B, unless C decides to complete
  - Occurrence of transaction from outsiders
- Allow decision-making
- Cause state change

EPMO Goals

- Agree on values:
  - C, M, B agree on each other's identities and price
  - C, M agree on goods; C, B agree on account number
- Preserve confidentiality: Protect
  - C's account number from M, outsiders
  - goods from B, outsiders
  - price from outsiders
  - M's identity from B, unless C decides to complete
  - Occurrence of transaction from outsiders
- Allow decision-making:
  - C decides to spend price for goods from M
  - M decides to sell goods to C for price
  - B decides to transfer price
- Cause state change

EPMO Goals

- Agree on values:
  - C, M, B agree on each other's identities and price
  - C, M agree on goods; C, B agree on account number
- Preserve confidentiality: Protect
  - C's account number from M, outsiders
  - goods from B, outsiders
  - price from outsiders
  - M's identity from B, unless C decides to complete
  - Occurrence of transaction from outsiders
- Allow decision-making:
  - C decides to spend price for goods from M
  - M decides to sell goods to C for price
  - B decides to transfer price
- Cause state change:
  - B transfers funds
  - M issues shipping order

Layers of analysis

- Protocol mechanics
- Trust management
- State and state change

Layers of analysis

- Protocol mechanics
  - Who sends what messages
  - Accounts for
    - Confidentiality
    - Authentication and agreement
- Trust management
- State and state change

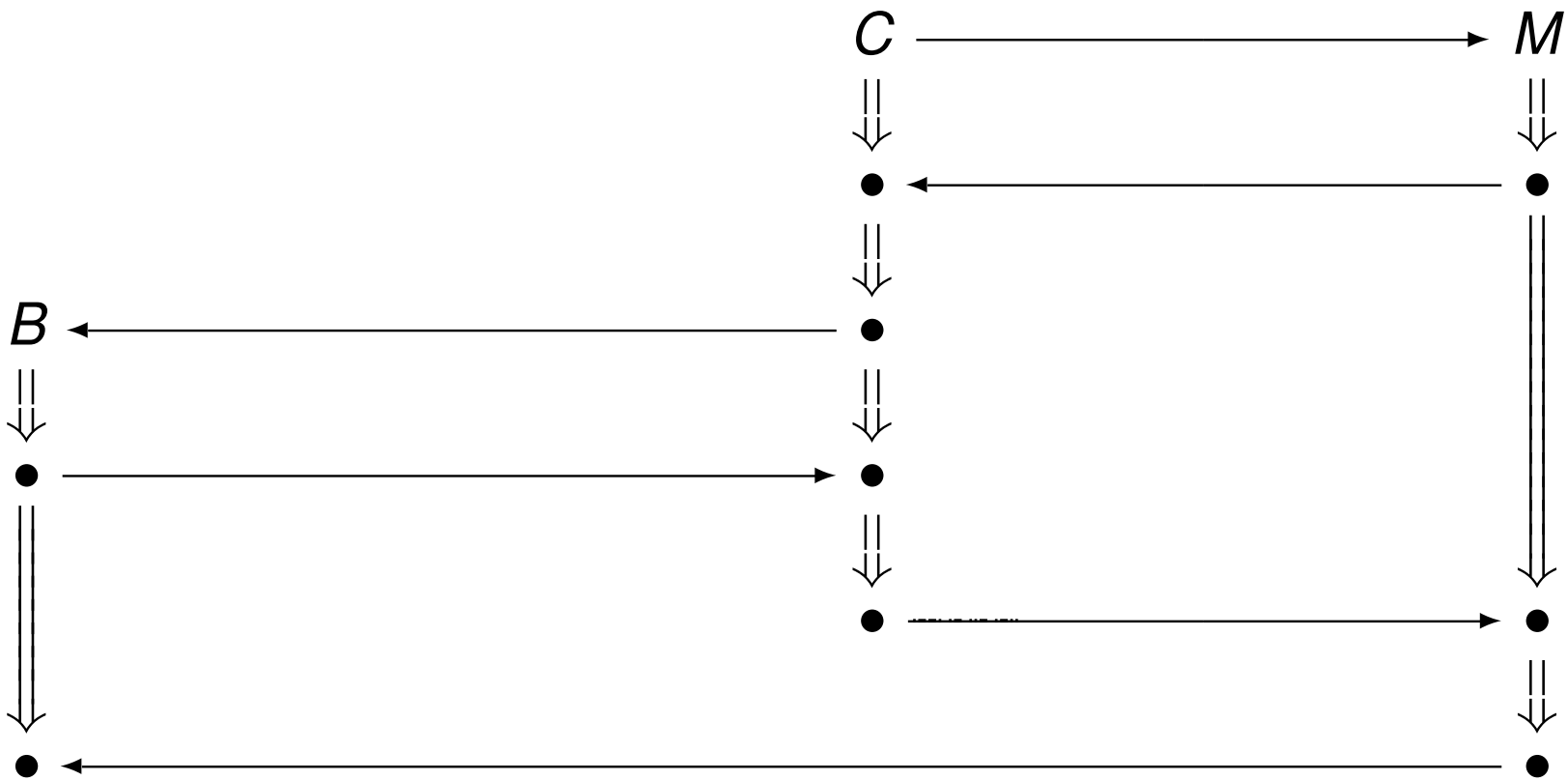
Layers of analysis

- Protocol mechanics
  - Who sends what messages
  - Accounts for
    - Confidentiality
    - Authentication and agreement
- Trust management
  - Decision making
  - Accounts for
    - Commitments made at each step
    - Protocol branching or early termination
- State and state change

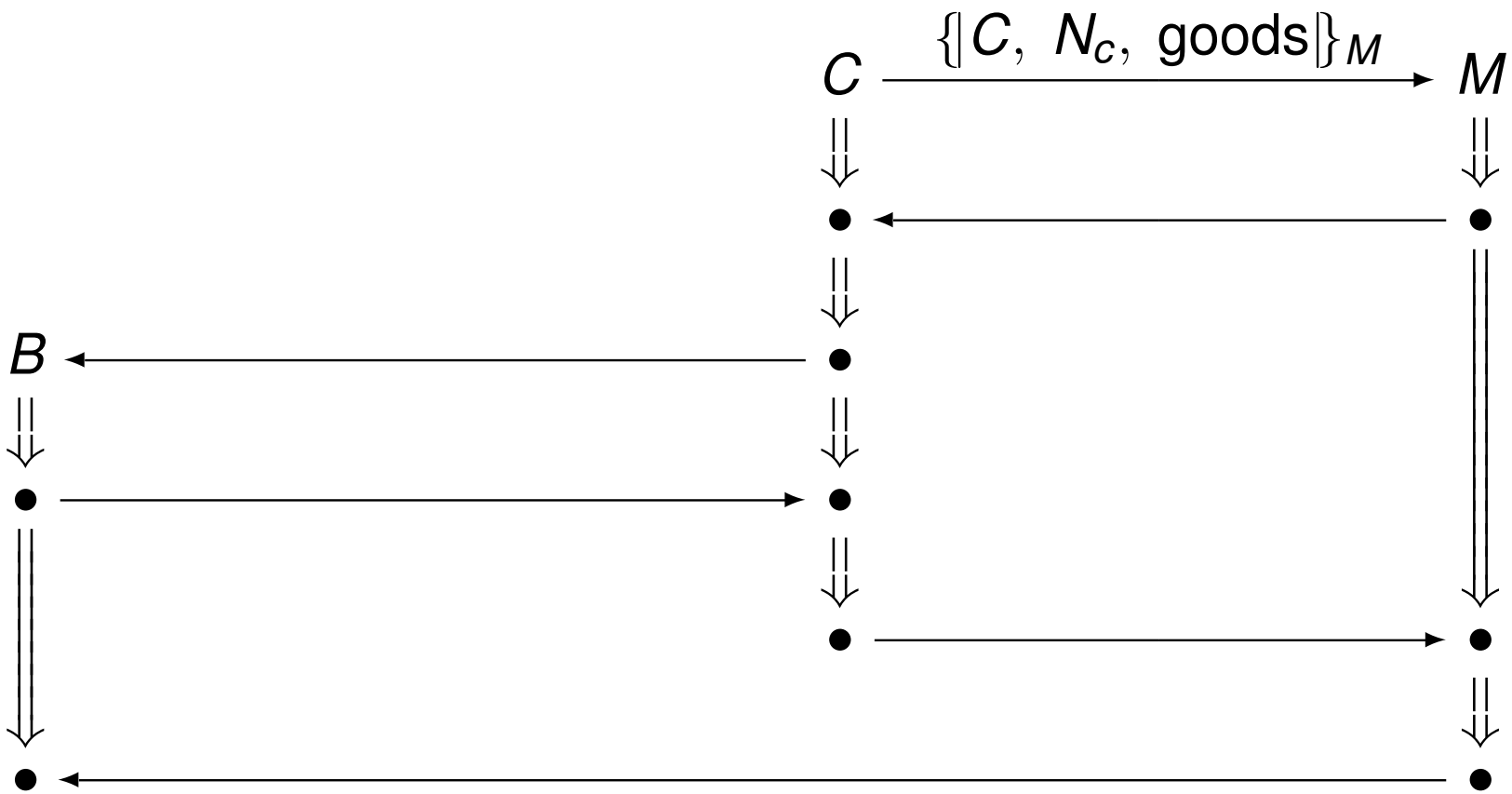
Layers of analysis

- Protocol mechanics
  - Who sends what messages
  - Accounts for
    - Confidentiality
    - Authentication and agreement
- Trust management
  - Decision making
  - Accounts for
    - Commitments made at each step
    - Protocol branching or early termination
- State and state change
  - Conditions and effects of the protocol run
  - Accounts for
    - Initial premises supplied to trust management
    - Changes induced by successful run

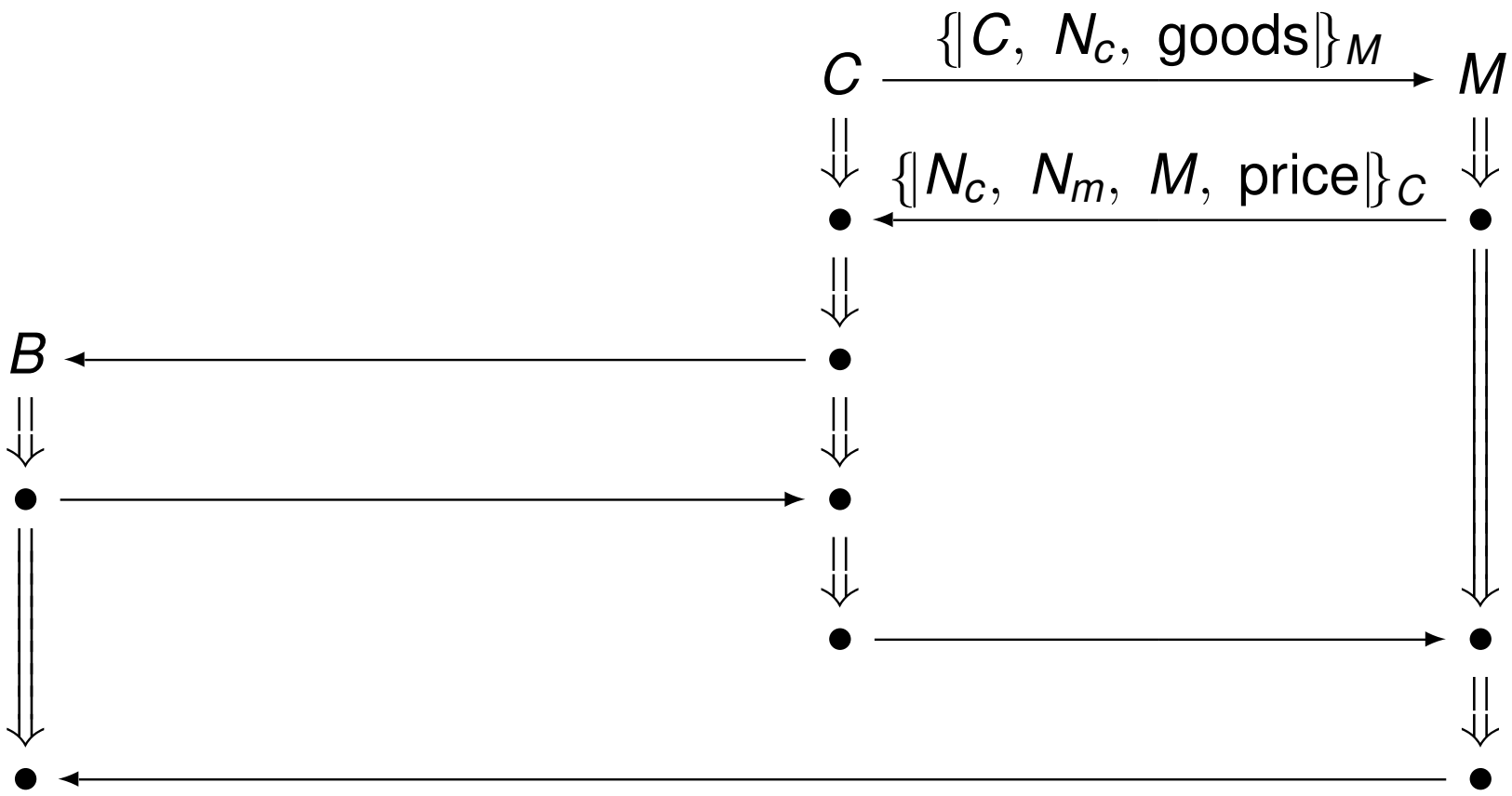
EPMO Protocol Structure, 1



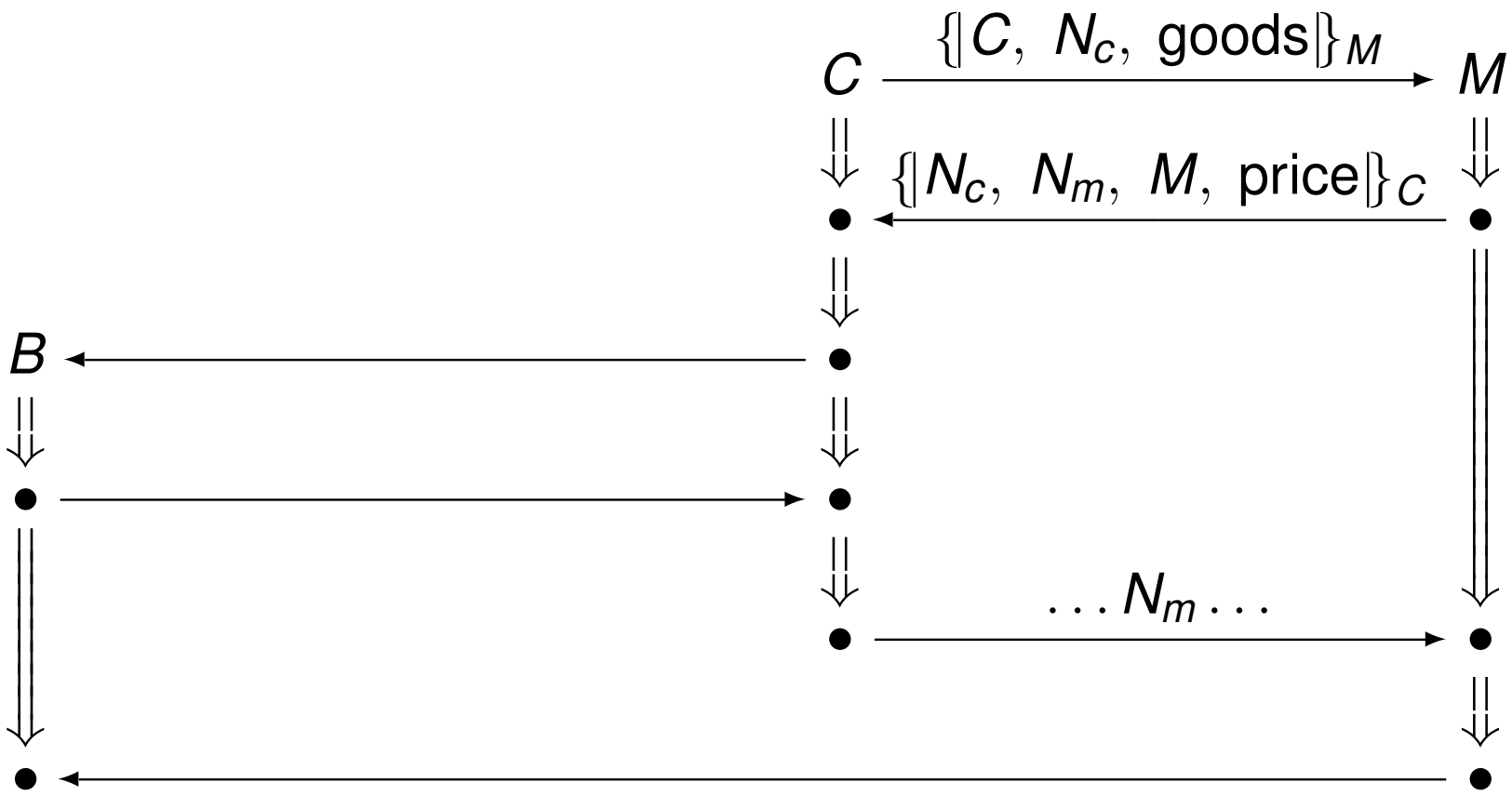
EPMO Protocol Structure, 1



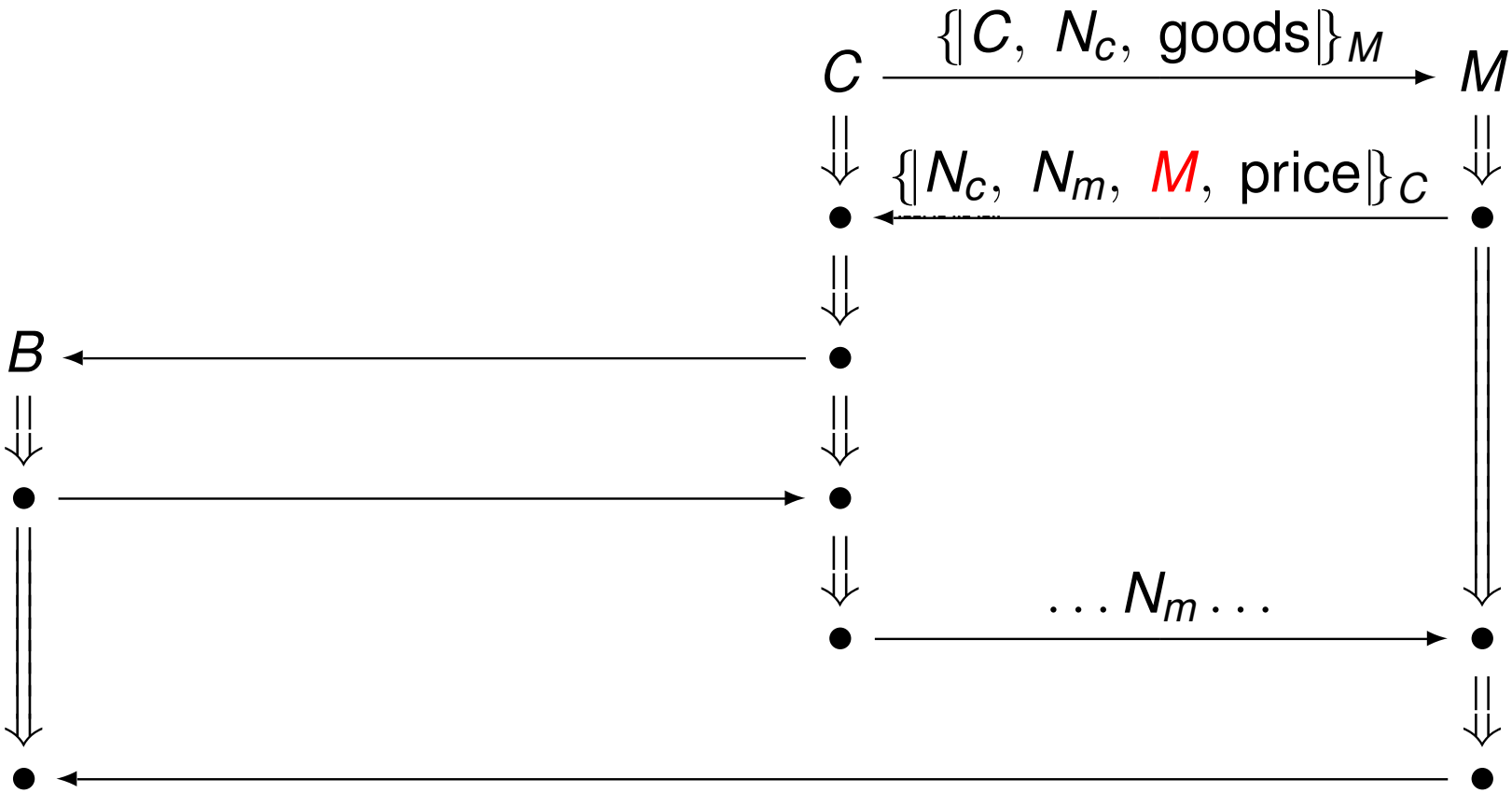
EPMO Protocol Structure, 1



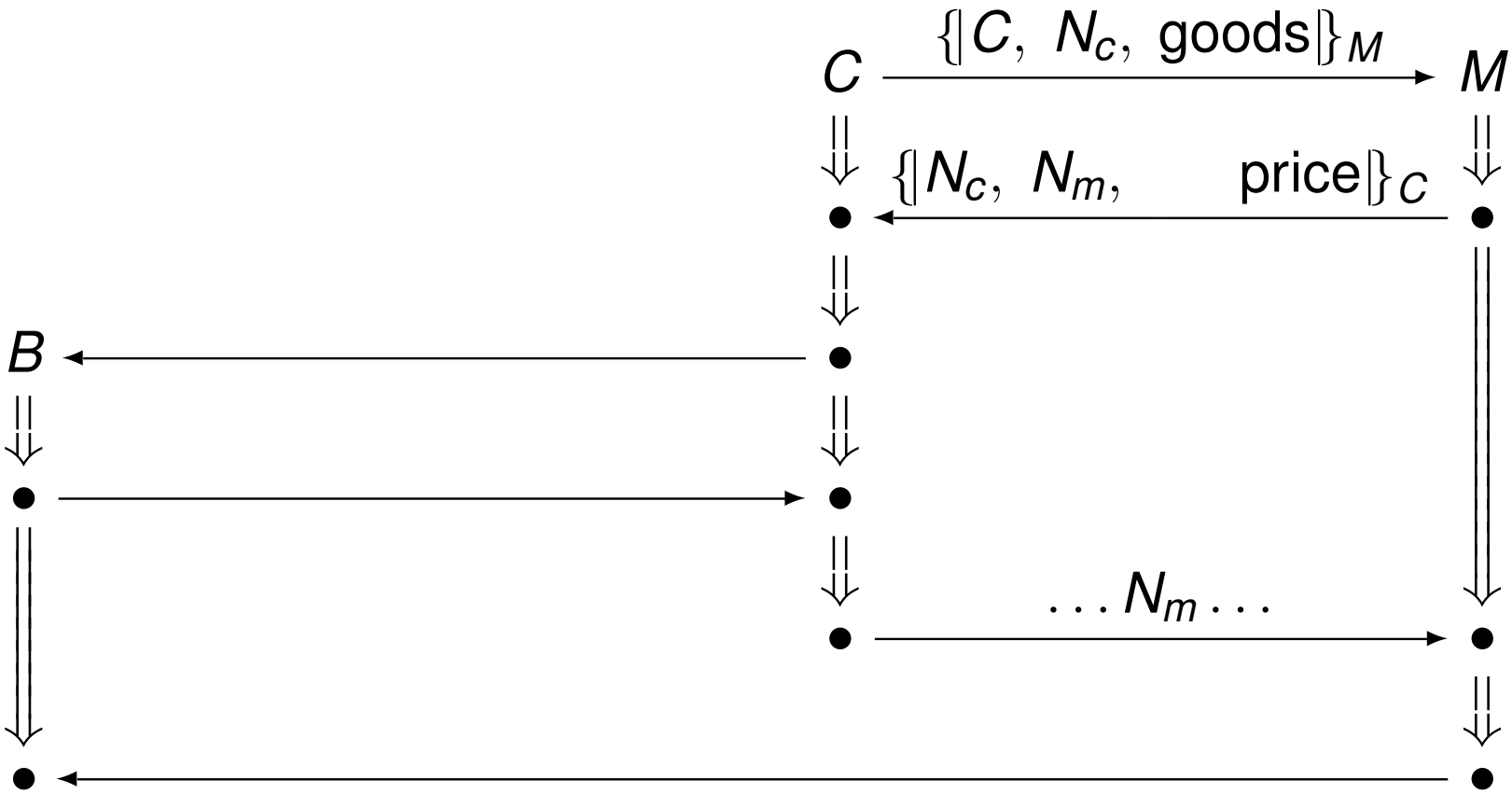
EPMO Protocol Structure, 1



EPMO Done Wrong

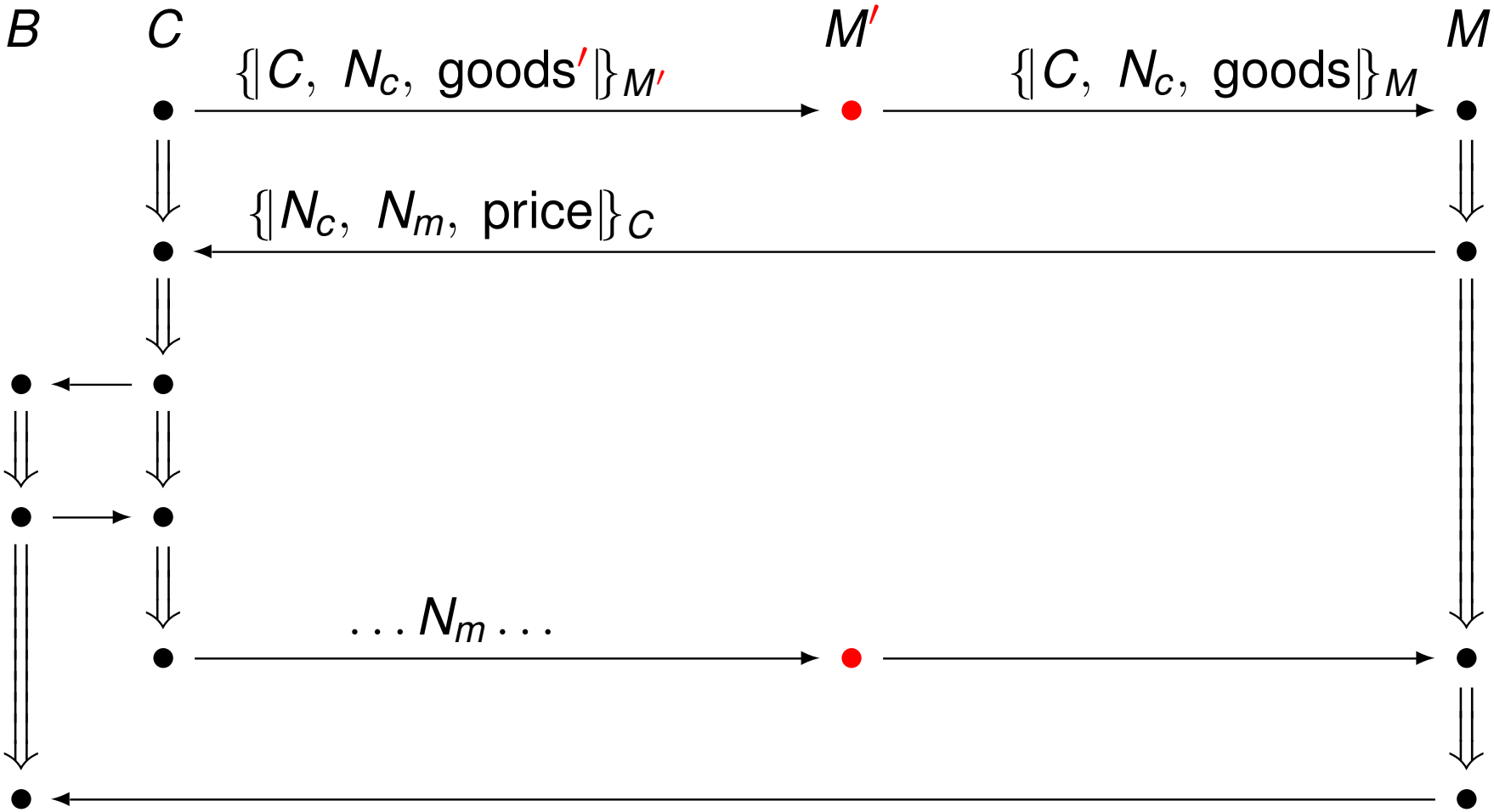


EPMO Done Wrong

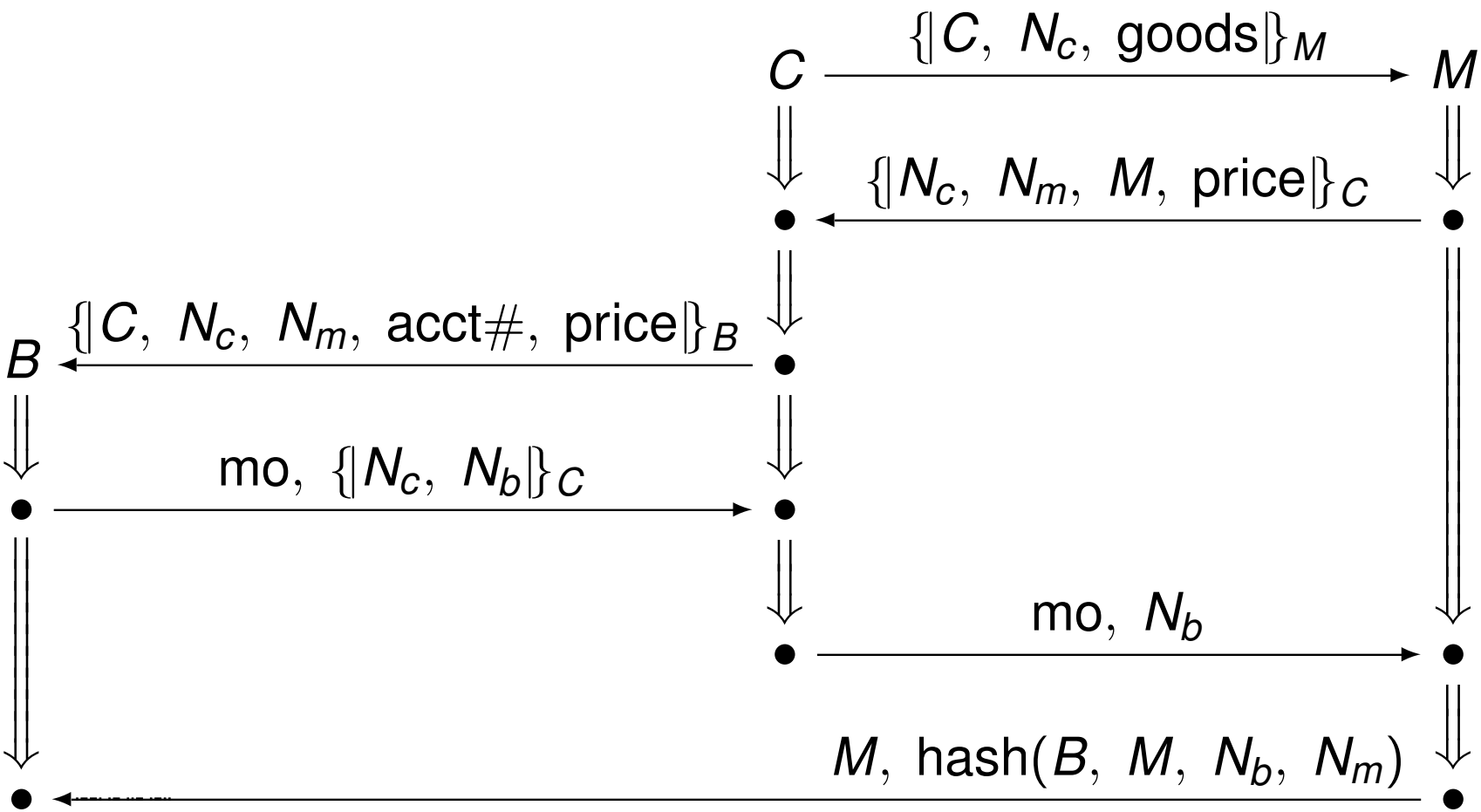


What if M's name omitted?

Lowe-style attack



EPMO Protocol Structure, 2



$$\mathrm{mo} = [\![\, \mathrm{hash}(C,\ N_c,\ N_b,\ N_m,\ \mathrm{price}) \,]\!]_B$$

Protocol Executions are Bundles

- Vertical columns are strands
  - Local, session-specific sequences
  - Could also represent adversary activity

Protocol Executions are Bundles

- Vertical columns are strands
  - Local, session-specific sequences
  - Could also represent adversary activity
- Transmissions, receptions on strands called “nodes”
  - Positive for send
  - Negative for receive

Protocol Executions are Bundles

- Vertical columns are strands
  - Local, session-specific sequences
  - Could also represent adversary activity
- Transmissions, receptions on strands called “nodes”
  - Positive for send
  - Negative for receive
- Bundle $\mathcal{B}$: causally well-founded execution
  Finite acyclic graph where
  - Every reception $-t$ has a unique transmission $+t$ where $+t \longrightarrow -t$
  - When $n_i \Rightarrow n_{i+1}$ on same strand, and n_{i+1} in $\mathcal{B}$, then n_i in $\mathcal{B}$

Protocol Executions are Bundles

- Vertical columns are strands
  - Local, session-specific sequences
  - Could also represent adversary activity
- Transmissions, receptions on strands called “nodes”
  - Positive for send
  - Negative for receive
- Bundle $\mathcal{B}$: causally well-founded execution
  Finite acyclic graph where
  - Every reception $-t$ has a unique transmission $+t$ where $+t \longrightarrow -t$
  - When $n_i \Rightarrow n_{i+1}$ on same strand, and n_{i+1} in $\mathcal{B}$, then n_i in $\mathcal{B}$

Edges are arrows: $\longrightarrow$ msg transmission arrows
$\Rightarrow$ strand succession arrows

Precedence within a Bundle

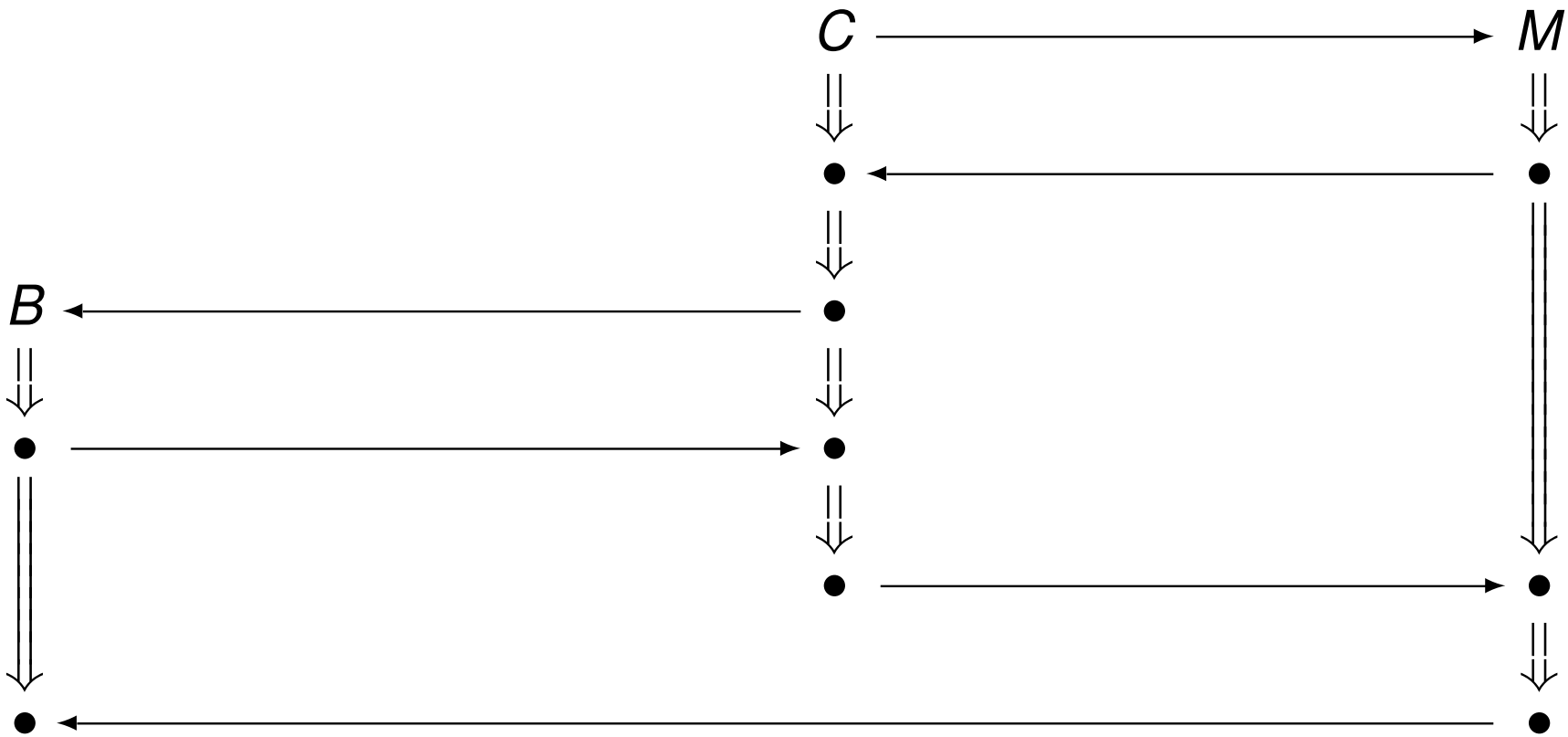
- Bundle precedence ordering $\preceq_{\mathcal{B}}$
  - $n \preceq_{\mathcal{B}} n'$ means sequence of 0 or more arrows $\rightarrow$, $\Rightarrow$ lead from n to n'
  - $\preceq_{\mathcal{B}}$ is a partial order by acyclicity
  - $\preceq_{\mathcal{B}}$ is well-founded by finiteness

Precedence within a Bundle

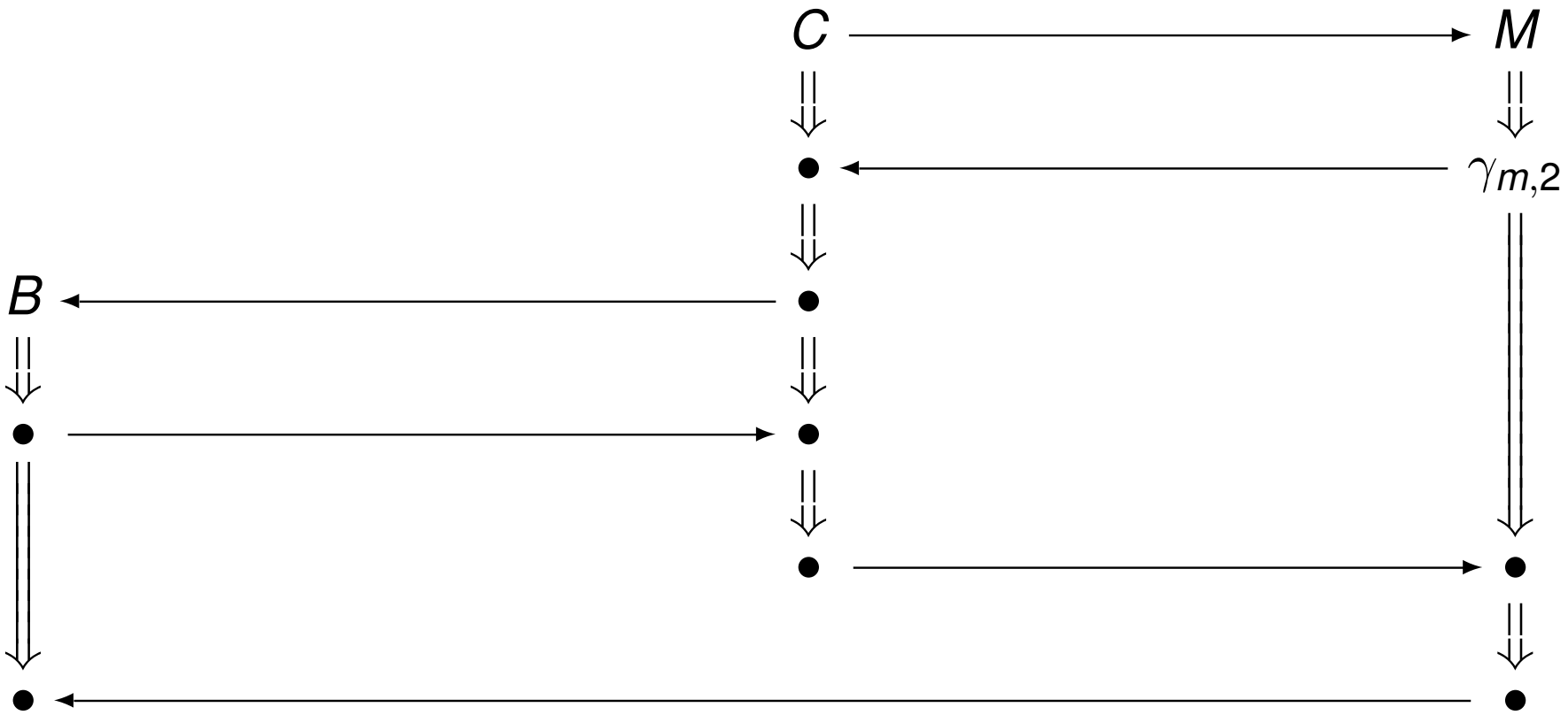
- Bundle precedence ordering $\preceq_{\mathcal{B}}$
  - $n \preceq_{\mathcal{B}} n'$ means sequence of 0 or more arrows $\rightarrow$, $\Rightarrow$ lead from n to n'
  - $\preceq_{\mathcal{B}}$ is a partial order by acyclicity
  - $\preceq_{\mathcal{B}}$ is well-founded by finiteness
- Bundle induction:

  Every non-empty subset of $\mathcal{B}$ has $\preceq_{\mathcal{B}}$-minimal members
- Reasoning about protocols combines
  - Bundle induction
  - Induction on message structure

Guarantees in Purchasing

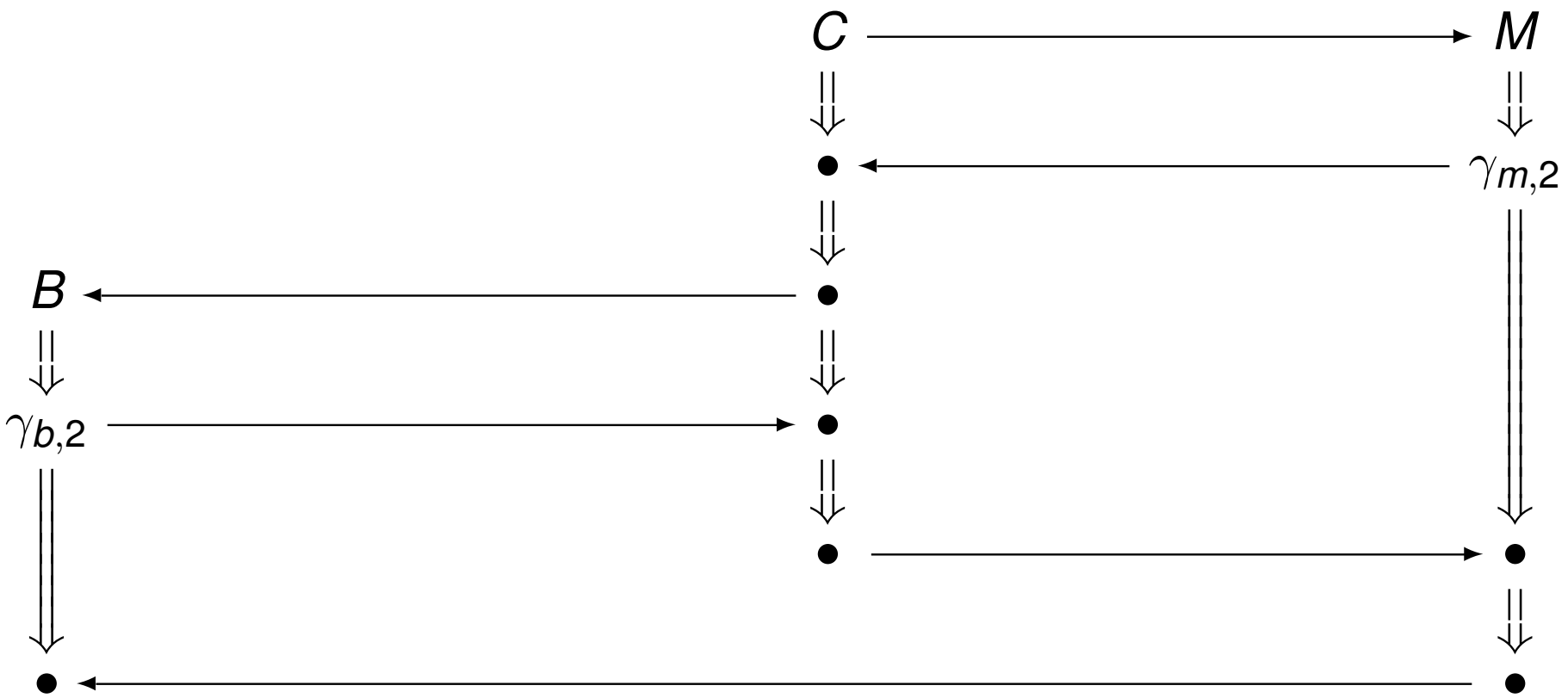


Guarantees in Purchasing



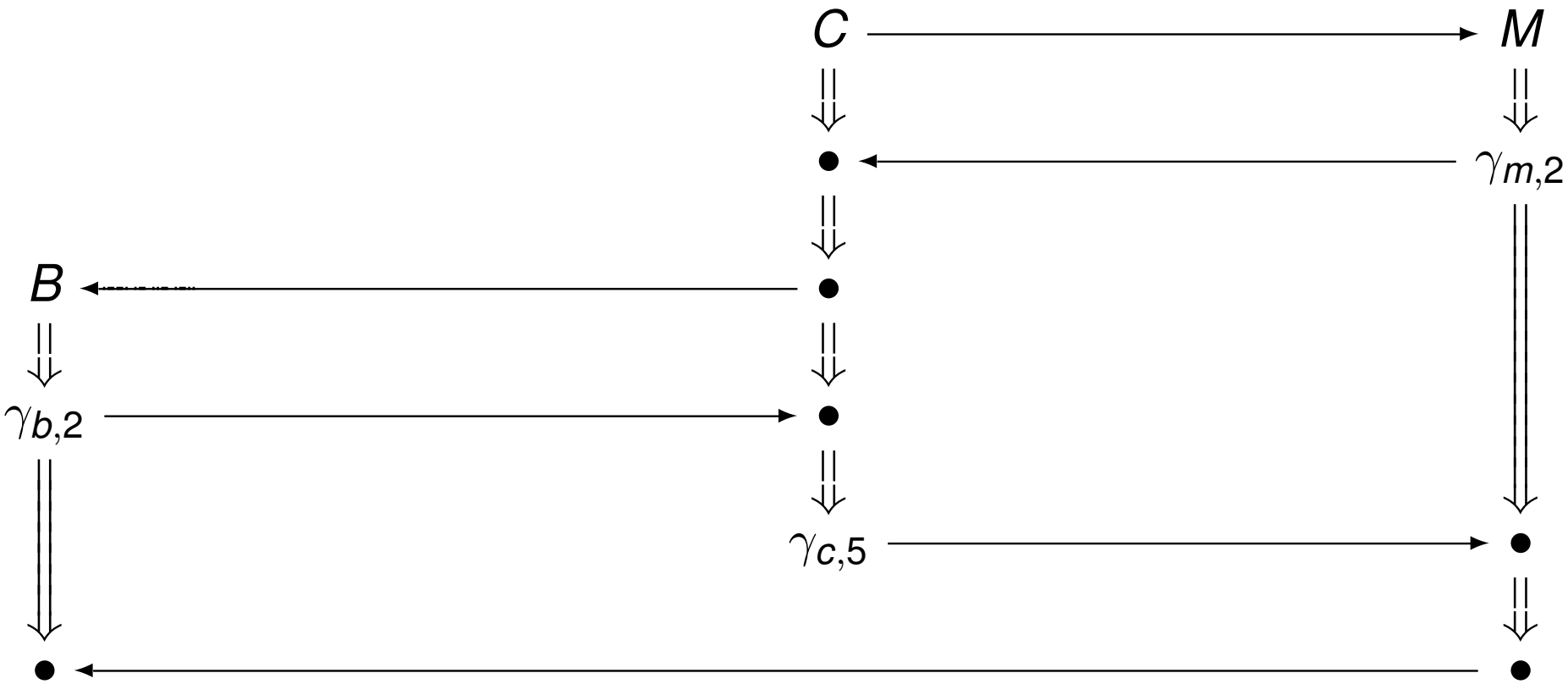
$\gamma_{m,2}$: I will ship C goods, if paid price

Guarantees in Purchasing



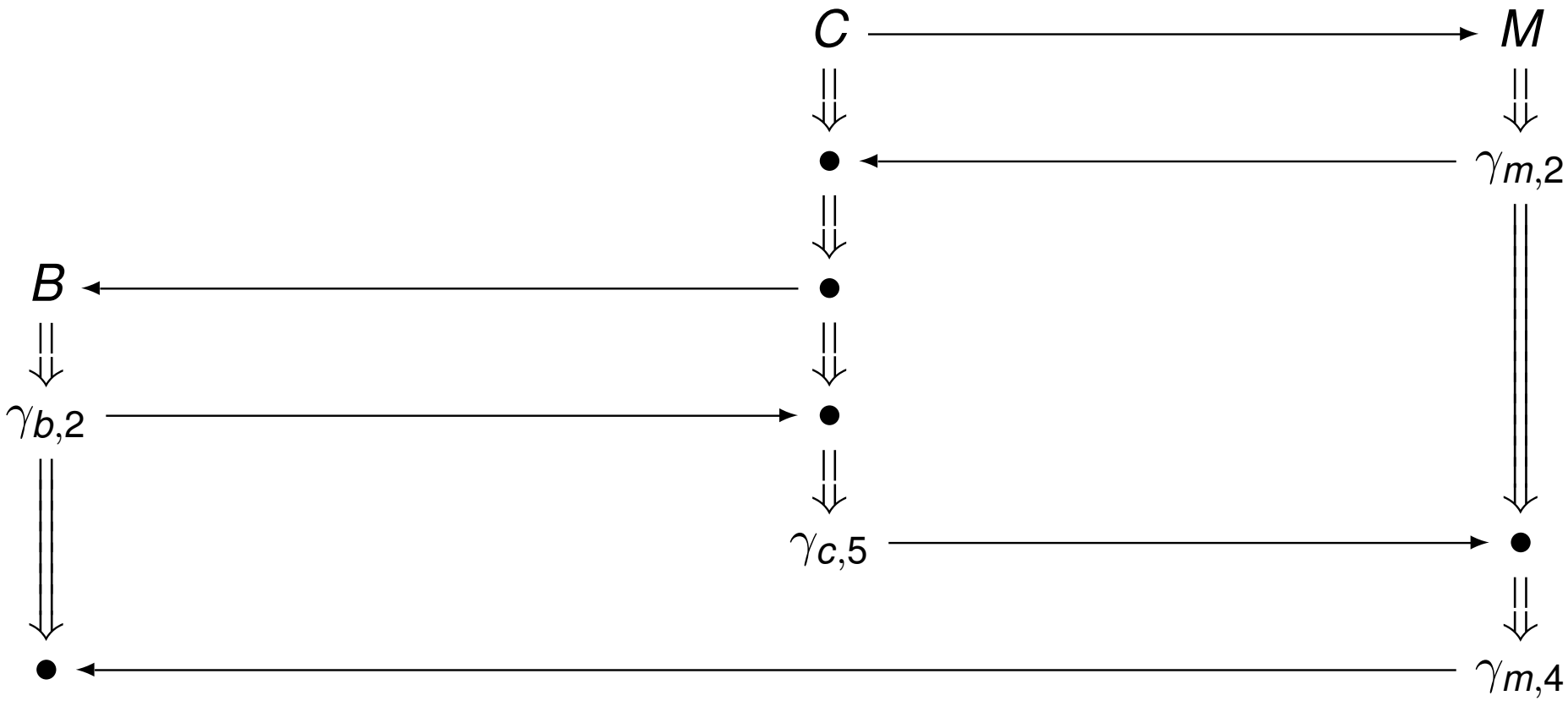
$\gamma_{b,2}$: I will pay price from acct# to the bearer P, if P authorized by C

Guarantees in Purchasing



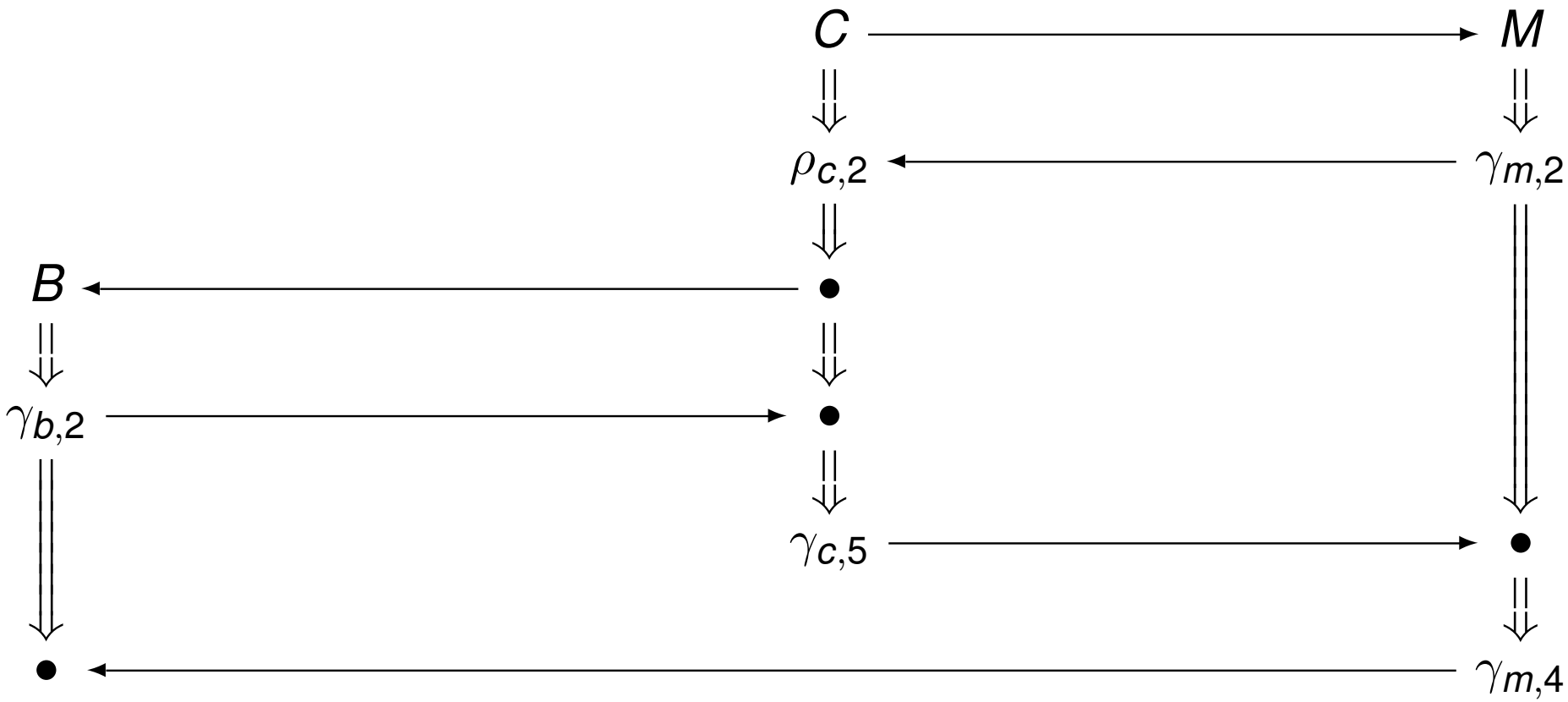
$\gamma_{c,5}$: I authorize payment from acct# to *M*

Guarantees in Purchasing



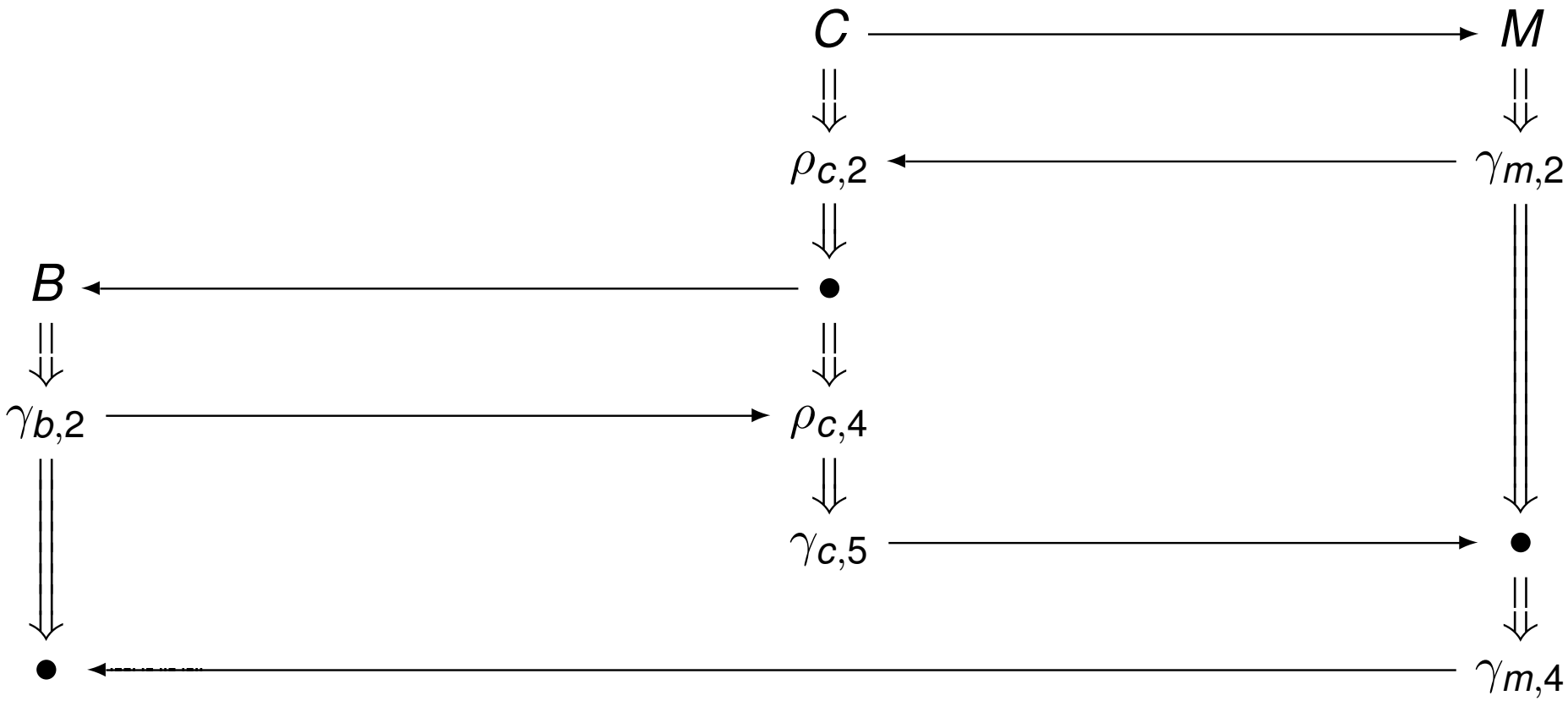
$\gamma_{m,4}$: I request payment and will ship C goods

Relying on others



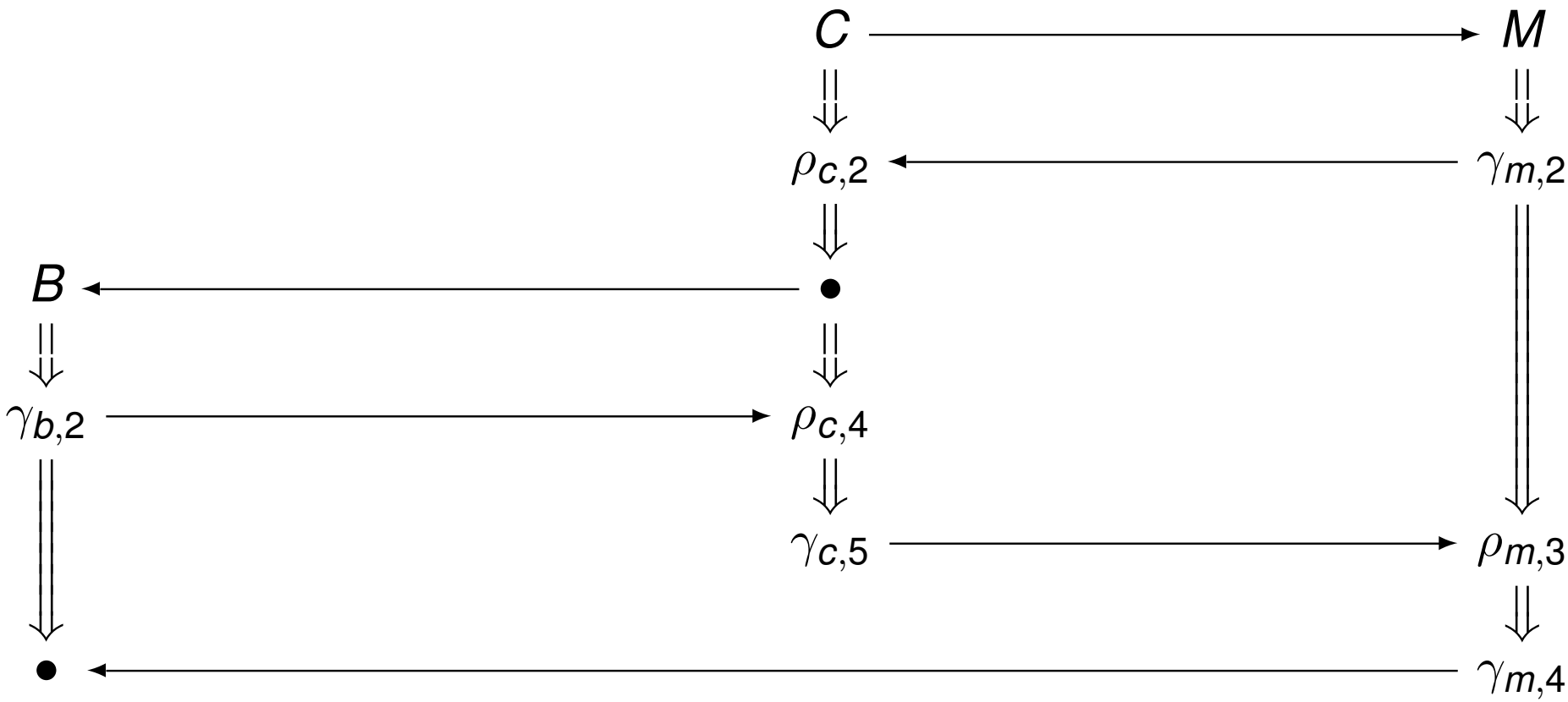
$\rho_{c,2}$: M says $\gamma_{m,2}$

Relying on others



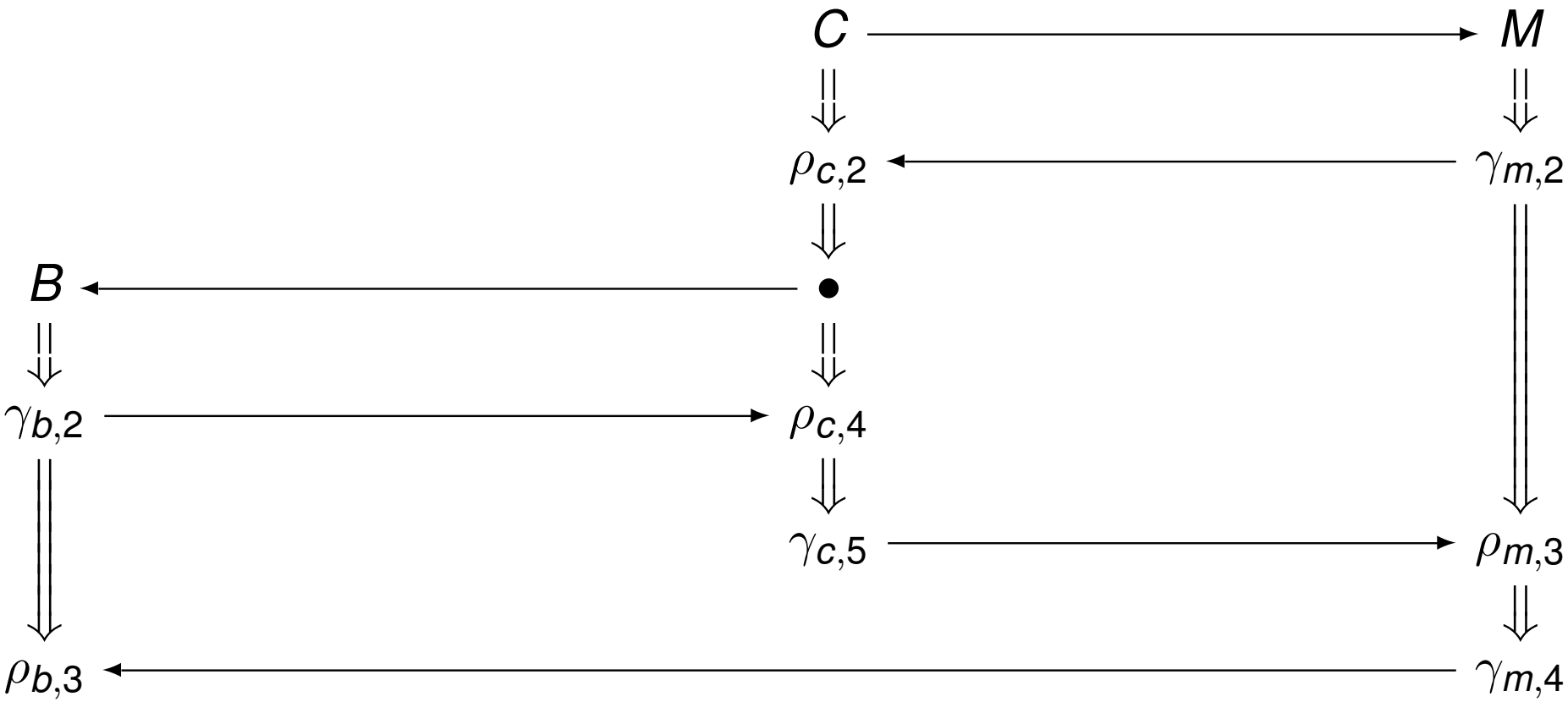
$\rho_{c,4}$: B says $\gamma_{b,2}$

Relying on others



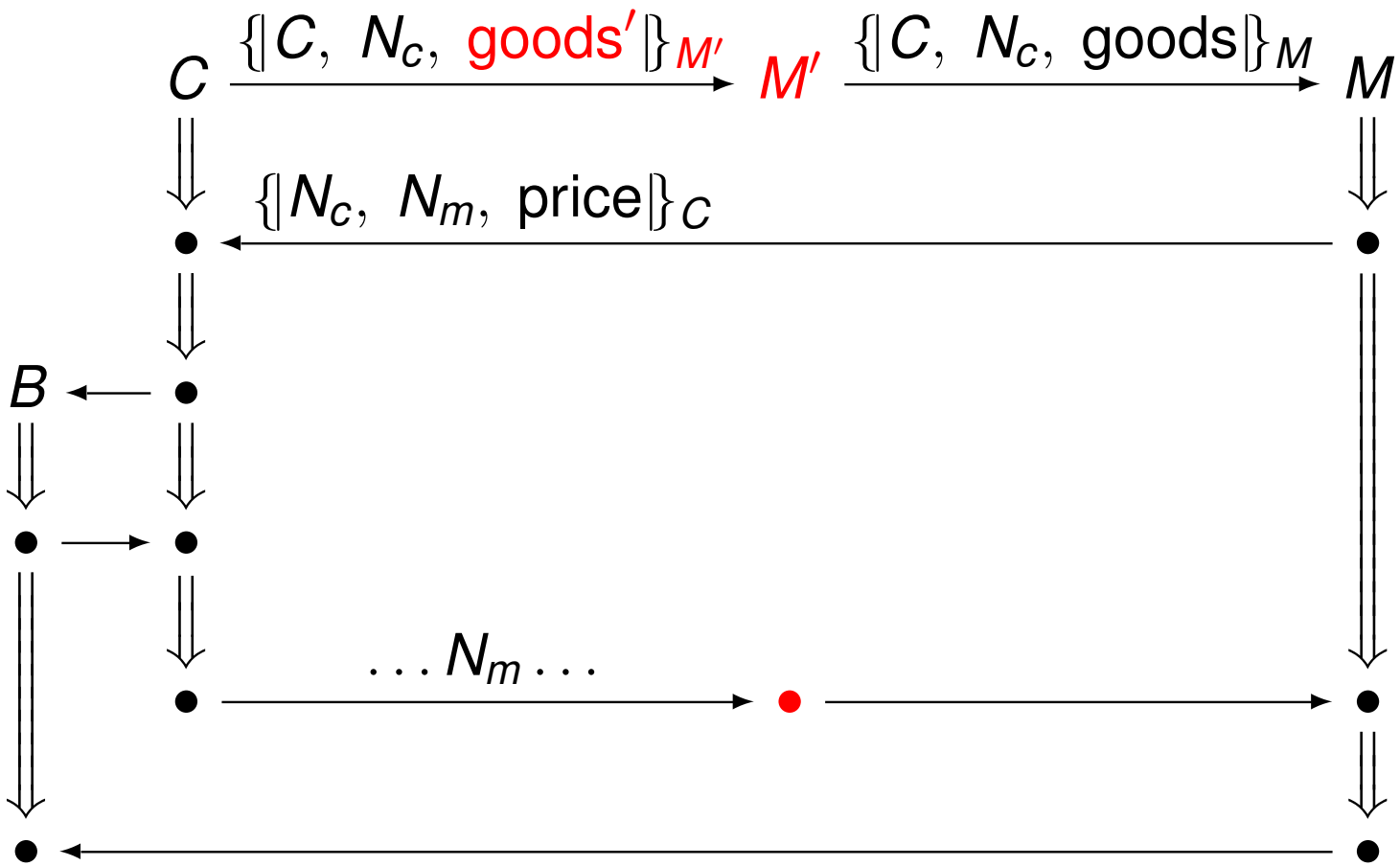
$\rho_{m,3}$: *B* says *B* will pay if authorized and
C says *C* authorizes payment to *M*

Relying on others



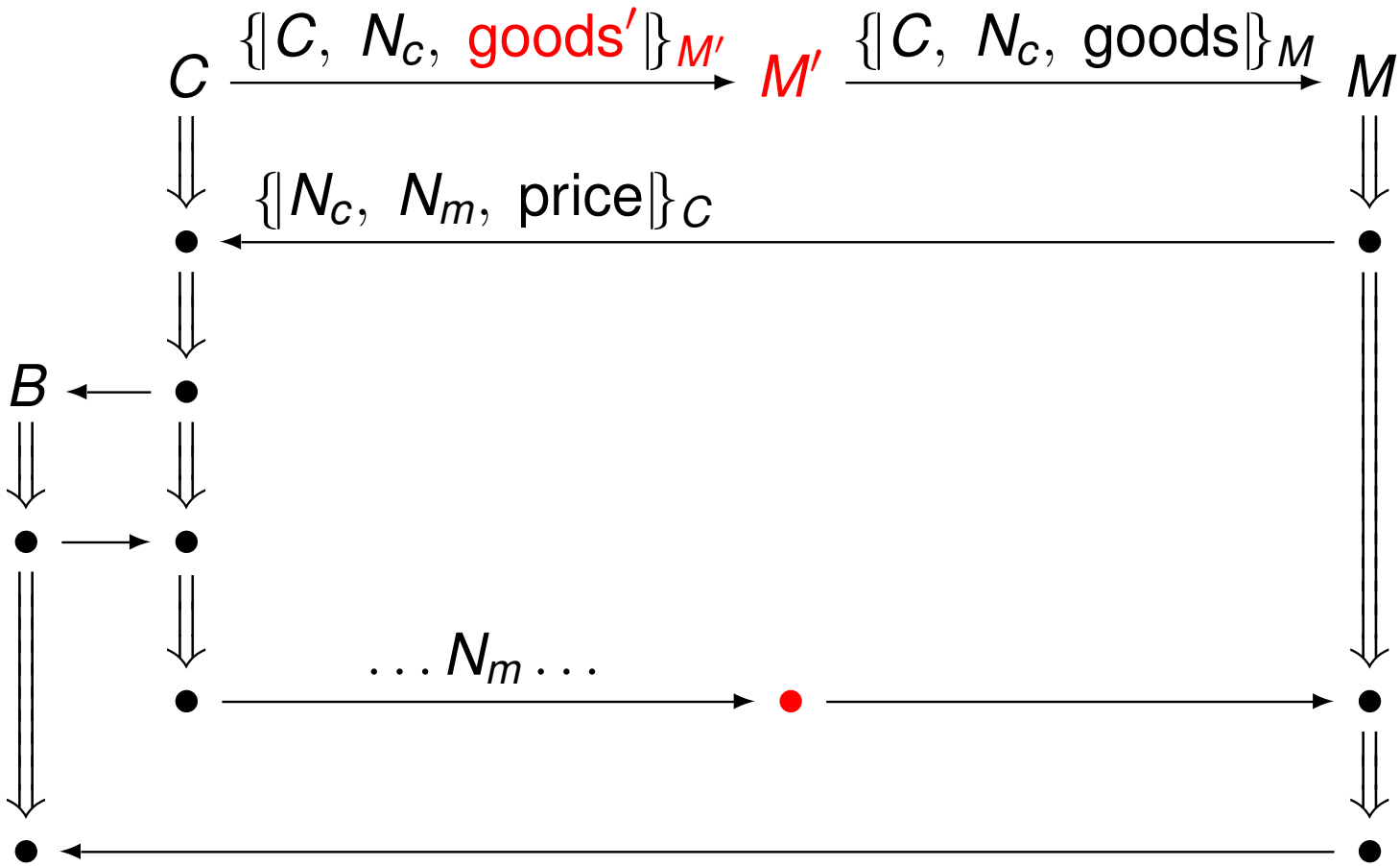
$\rho_{b,3}$: C says C authorizes payment from acct# to M and M says M requests payment

Lowe-style attack



$\gamma_{c,5}$: I authorize payment from acct# to M'

Lowe-style attack



$\rho_{m,3}$: B says B will pay if authorized and
C says C authorizes payment to M

Protocol Soundness

- Protocol Π is sound iff, for all executions $\mathcal{B}$ of Π, and message receptions $n \in \mathcal{B}$

$$\{\text{prin}(m) \text{ says } \gamma_m : m \prec_{\mathcal{B}} n\} \quad \longrightarrow_{\mathcal{L}} \quad \rho_n$$

  where

  $\longrightarrow_{\mathcal{L}}$ is the consequence relation of the underlying logic

  $\prec_{\mathcal{B}}$ is the bundle partial order

- Soundness follows from authentication properties
  - Strand space authentication methods work fine
  - Recency easy to incorporate
- Soundness means Π strong enough for its trust interpretation

Trust and State

- EPMO needs to consult state

  M must know goods in stock;
  price is right

  B must know price available in acct#

- EPMO needs to modify state when successful

  M needs to decrement inventory;
  produce shipping order

  B needs to transfer price from *C* to *M*

Events involving state

Bank run with events shown:

$$\bullet \Longrightarrow \mathsf{E\_hold}(\ldots) \Longrightarrow \bullet \Longrightarrow \bullet \Longrightarrow \mathsf{E\_transfer}(\ldots)$$

Maintaining state

- Multiset rewriting
  - A simple and appealing formalism
- Represents state as multiset of facts
- Rewrite rule $\rho =$

$$E(\vec{t_0}), \ldots, F(\vec{t_1}) \rightarrow G(\vec{t_2}), \ldots, H(\vec{t_3})$$

consumes facts matching $E(\vec{t_0}), \ldots, F(\vec{t_1})$,
produces facts matching $G(\vec{t_2}), \ldots, H(\vec{t_3})$

Maintaining state

- Multiset rewriting
  - A simple and appealing formalism
- Represents state as multiset of facts
- Rewrite rule $\rho =$

$$E(\vec{t_0}), \ldots, F(\vec{t_1}) \rightarrow G(\vec{t_2}), \ldots, H(\vec{t_3})$$

  consumes facts matching $E(\vec{t_0}), \ldots, F(\vec{t_1})$,
  produces facts matching $G(\vec{t_2}), \ldots, H(\vec{t_3})$
- Assume all variables in $\vec{t_2}, \ldots, \vec{t_3}$ appear in $\vec{t_0}, \ldots, \vec{t_1}$

Applying $\rho = E(\vec{t_0}), \ldots, F(\vec{t_1}) \longrightarrow G(\vec{t_2}), \ldots, H(\vec{t_3})$

- Rule ρ is applicable to state Σ_0 if $\Sigma_0 =$

$$\Sigma_1, E(\vec{t_0} \cdot \sigma), \ldots, F(\vec{t_1} \cdot \sigma)$$

- Produces transition of form

$$\begin{array}{rl} & \Sigma_1, E(\vec{t_0} \cdot \sigma), \ldots, F(\vec{t_1} \cdot \sigma) \\ \overset{\rho,\sigma}{\longmapsto} & \Sigma_1, G(\vec{t_2} \cdot \sigma), \ldots, H(\vec{t_3} \cdot \sigma) \end{array}$$

- "Computation" means path

$$\mathcal{C} = \Sigma_0 \overset{\rho_0,\sigma_0}{\longmapsto} \Sigma_1 \overset{\rho_1,\sigma_1}{\longmapsto} \Sigma_2 \ldots \overset{\rho_2,\sigma_2}{\longmapsto}$$

A convention of use: Locality

We assume

1. Every predicate is of the form $F(p, \vec{t})$ where p is a principal
2. Every rule takes form $\rho =$

$$E(p, \vec{t_0}), \ldots, F(p, \vec{t_1}) \rightarrow G(p, \vec{t_2}), \ldots, H(p, \vec{t_3})$$

Every computation step affects a single principal p

Consequence of locality: Concurrency

- Events on different principals are always concurrent
- If $p \cdot \sigma_1 \neq p \cdot \sigma_2$ and $(\rho_1, \sigma_1), (\rho_2, \sigma_2)$ can happen, so can reverse:

$$\Sigma_0 \overset{\rho_1,\sigma_1}{\longmapsto} \Sigma_1 \overset{\rho_2,\sigma_2}{\longmapsto} \Sigma_2$$

  implies

$$\Sigma_0 \overset{\rho_2,\sigma_2}{\longmapsto} \Sigma_1' \overset{\rho_1,\sigma_1}{\longmapsto} \Sigma_2$$

Consequence of locality: Concurrency

- Events on different principals are always concurrent
- If $p \cdot \sigma_1 \neq p \cdot \sigma_2$ and $(\rho_1, \sigma_1), (\rho_2, \sigma_2)$ can happen, so can reverse:

  $$\Sigma_0 \overset{\rho_1,\sigma_1}{\longmapsto} \Sigma_1 \overset{\rho_2,\sigma_2}{\longmapsto} \Sigma_2$$

  implies

  $$\Sigma_0 \overset{\rho_2,\sigma_2}{\longmapsto} \Sigma_1' \overset{\rho_1,\sigma_1}{\longmapsto} \Sigma_2$$

- Rules never coordinate different principals; only protocols coordinate principals

Second convention of use: Events

We assume

1. Finite set of predicates $\{E_i(p, \vec{t_i})\}_{i \in I}$ is distinguished, called the events
2. Each rule ρ either produces one event:

$$F_1(p, \vec{t_0}), \ldots, F_k(p, \vec{t_1}) \rightarrow E_i(p, \vec{t_2})$$

or consumes one event:

$$E_i(p, \vec{t_0}) \rightarrow G_1(p, \vec{t_1}), \ldots, G_\ell(p, \vec{t_2})$$

where $F_1, \ldots, F_k, G_1, \ldots G_\ell$ are not events

Second convention of use: Events

We assume

1. Finite set of predicates $\{E_i(p, \vec{t_i})\}_{i \in I}$ is distinguished, called the events
2. Each rule ρ either produces one event:

$$F_1(p, \vec{t_0}), \ldots, F_k(p, \vec{t_1}) \rightarrow E_i(p, \vec{t_2})$$

or consumes one event:

$$E_i(p, \vec{t_0}) \rightarrow G_1(p, \vec{t_1}), \ldots, G_\ell(p, \vec{t_2})$$

where $F_1, \ldots, F_k, G_1, \ldots G_\ell$ are not events

Every computation equivalent to one in which every event is immediately consumed

Redefine strand traces

- Add events to strands
- The trace tr(s) is a sequence of nodes of three kinds:
  1. Message transmissions
  2. Message receptions
  3. Events $E_i(p, \vec{t})$

Compatible bundles and computations

Bundle $\mathcal{B}$ is compatible with computation $\mathcal{C}$ iff

there is a bijection ϕ from

- *Event nodes n of $\mathcal{B}$ to*
- *Event consumption steps ρ, σ of $\mathcal{C}$*

such that

- *The event $E_i(p, \vec{t})$ at n is the event consumed on $\phi(n)$*
- *$n_0 \preceq_{\mathcal{B}} n_1$ implies $\phi(n_0)$ precedes $\phi(n_1)$ in $\mathcal{C}$*

Execution model

$(\mathcal{B}, \mathcal{C})$ is an execution
for a protocol and a set of rules iff

- $\mathcal{B}$ is a bundle for the protocol
- $\mathcal{C}$ is a computation for the rules
- $\mathcal{B}$ and $\mathcal{C}$ are compatible

Protocols coordinate distributed systems

Protocols allow principals

- To agree on values
  - While preserving their secrecy
- To accept, select, or refuse commitments
  Often dependent on
  - Commitments received
  - Current local state
- To coordinate state changes
  between themselves

Despite presence of malicious adversaries