# CONTROLLERS FOR VERIFICATION OF DISTRIBUTED SYSTEMS

**Aiswarya Cyriac**
Uppsala University, Sweden

Joint work with

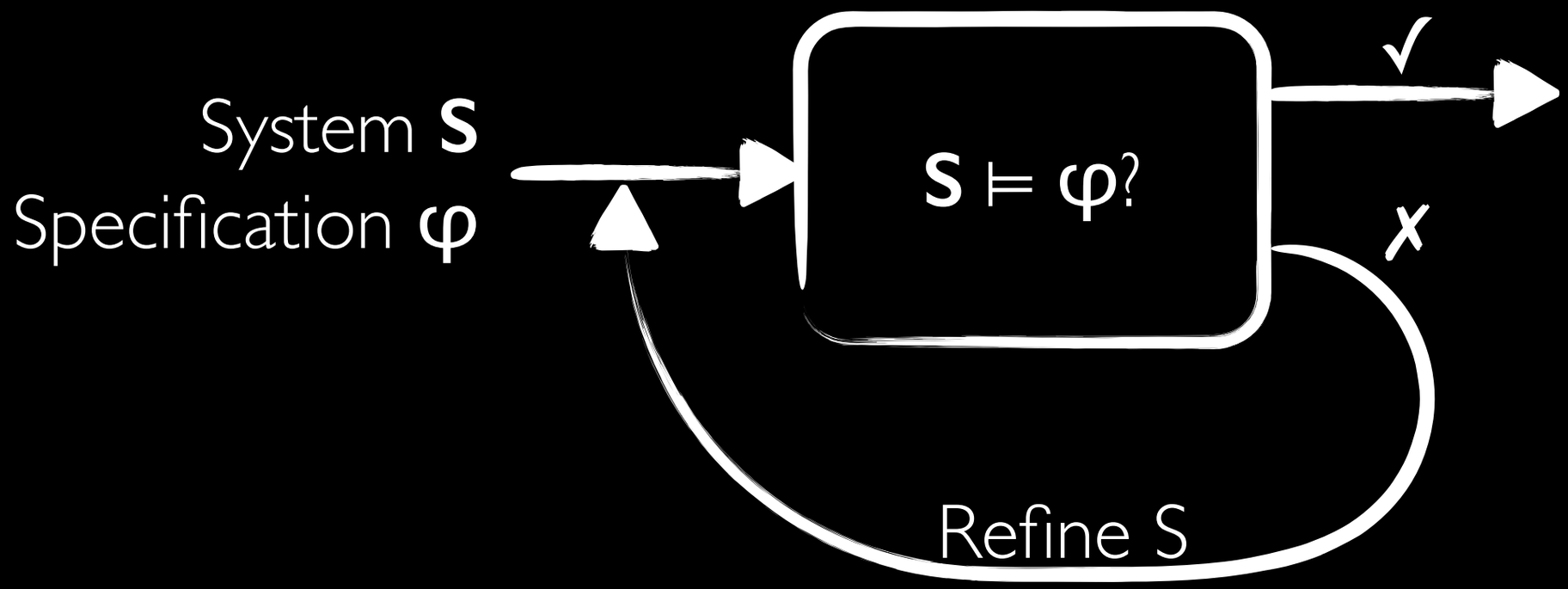
**Paul Gastin**
LSV, ENS Cachan, France

**K. Narayan Kumar**
Chennai Mathematical Institute, India

**Journées nationales du GDR-IM**
**02/02/2015 - Bordeaux**

# VERIFICATION

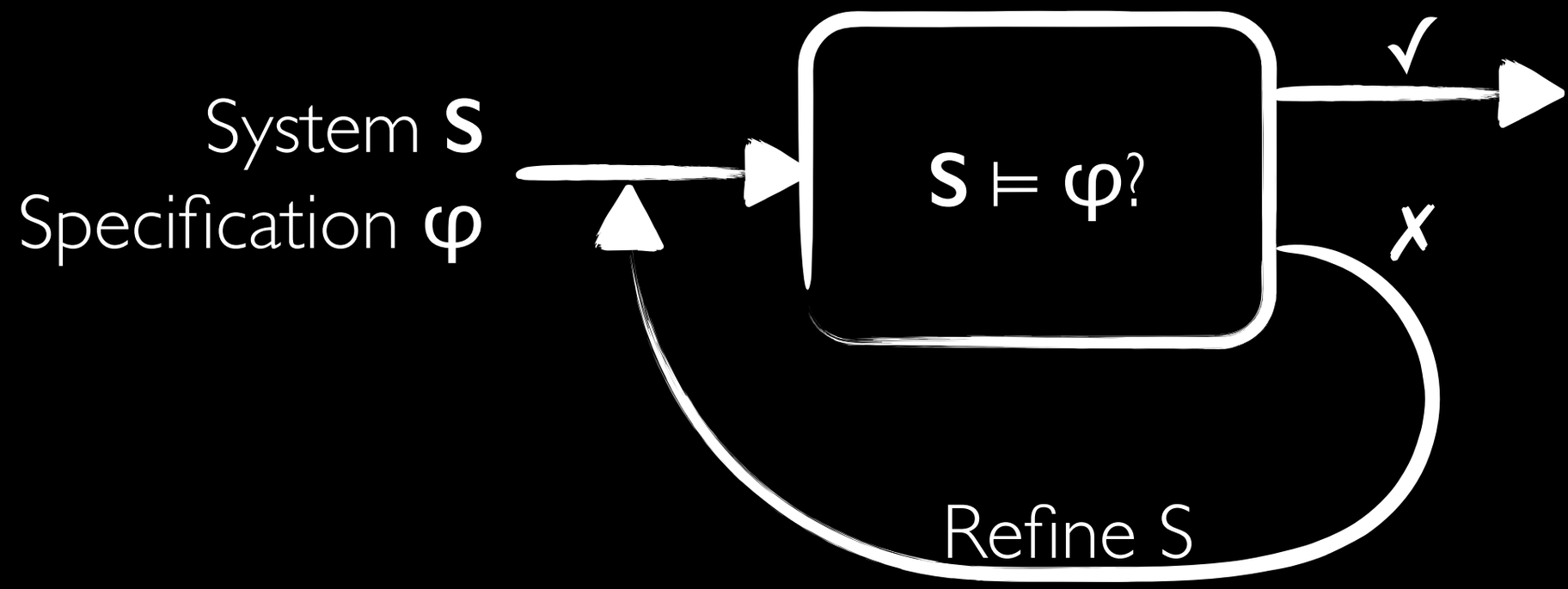
## Model Checking



# VERIFICATION

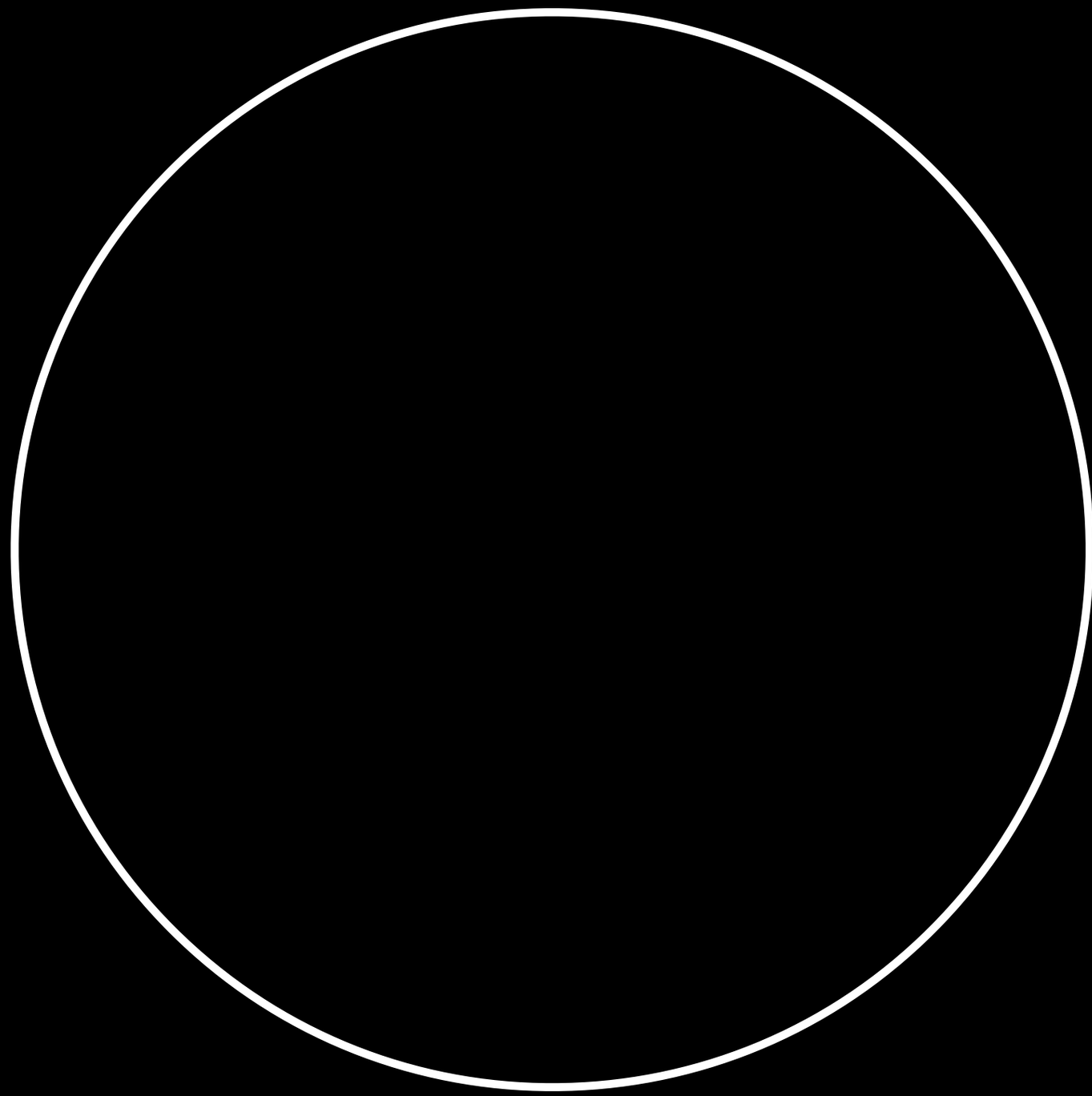
**Model Checking**

> **Undecidable in many cases**



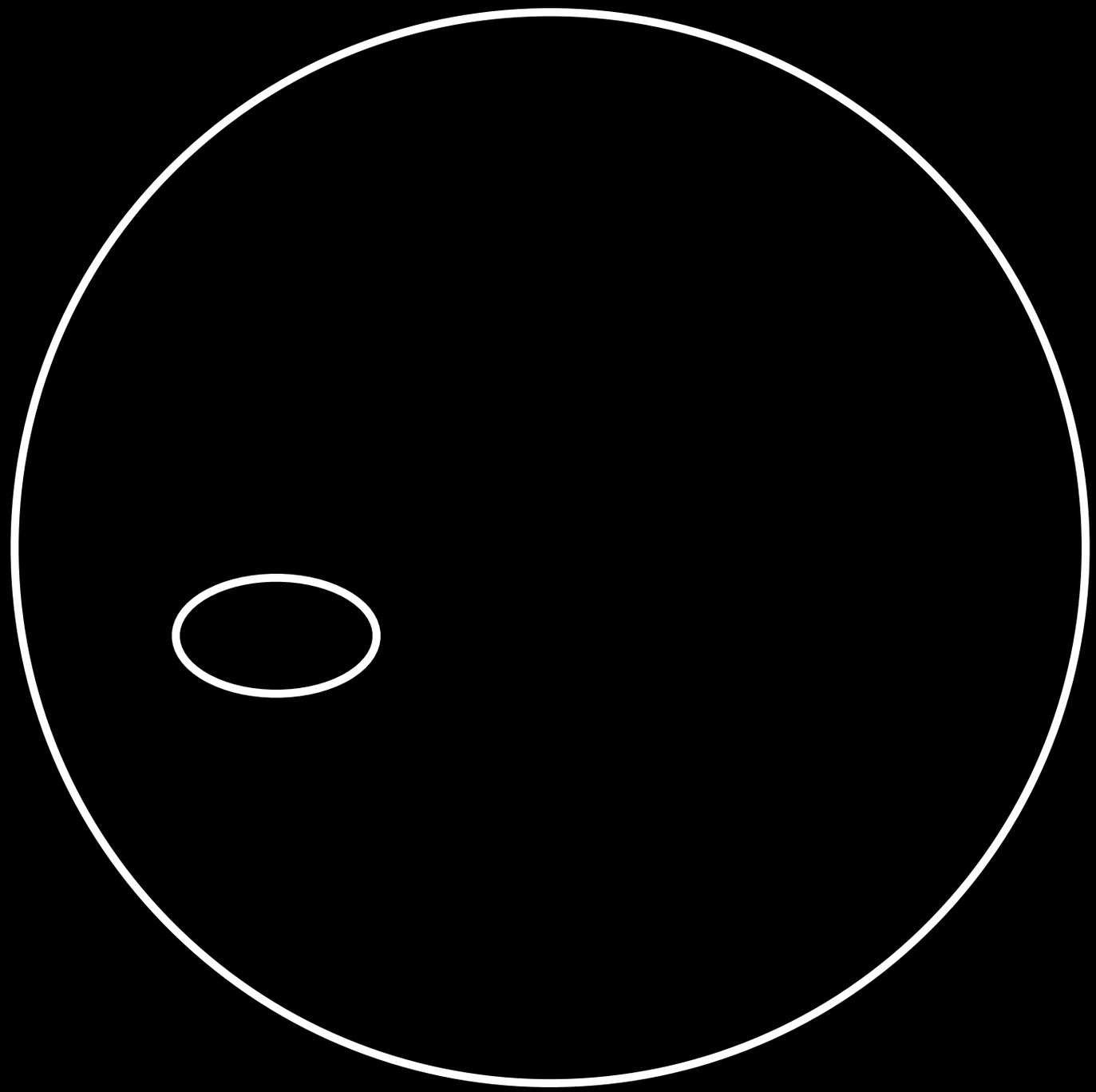
# UNDER-APPROXIMATE VERIFICATION

- Parametrised



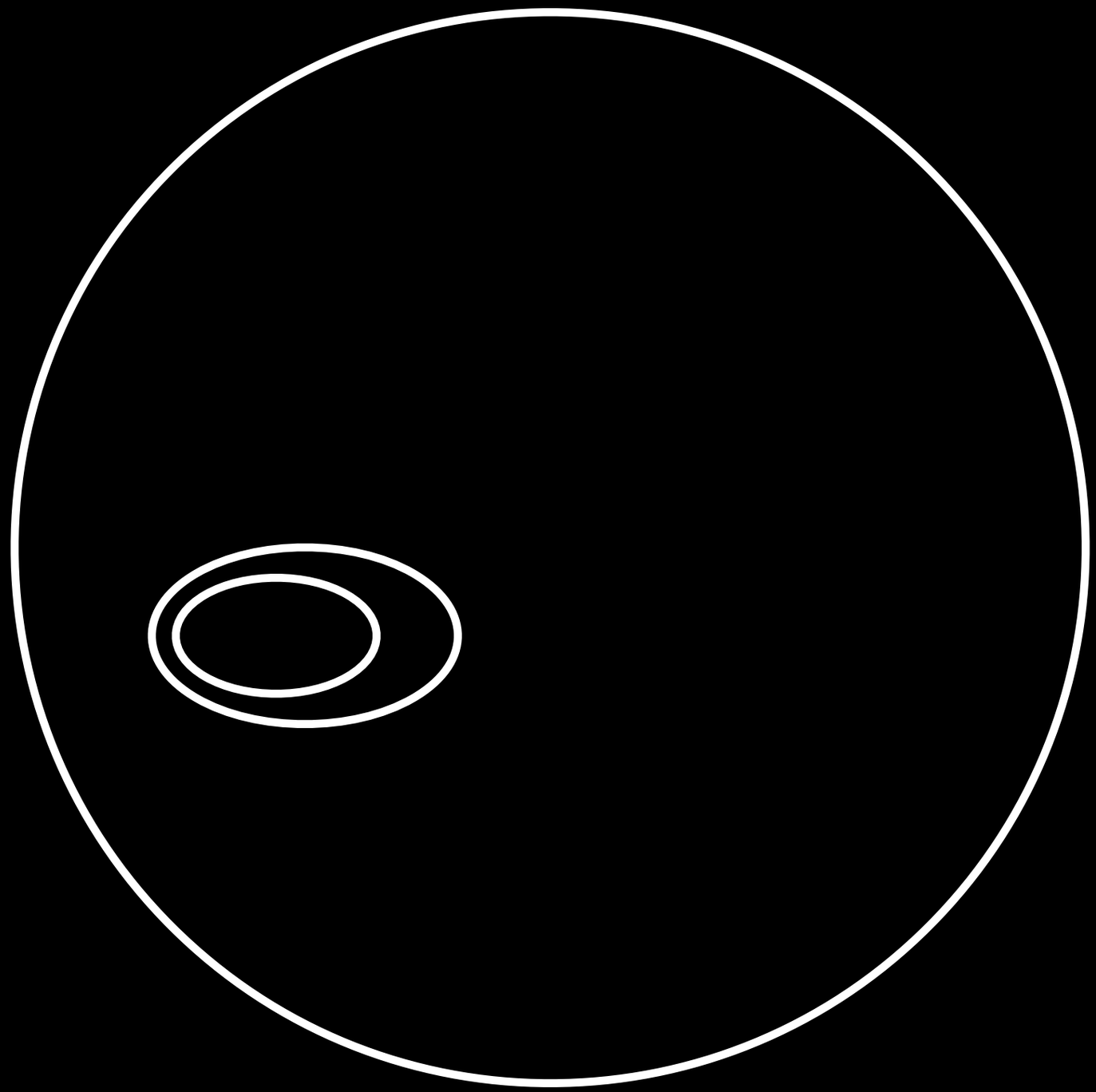
# UNDER-APPROXIMATE VERIFICATION

- Parametrised



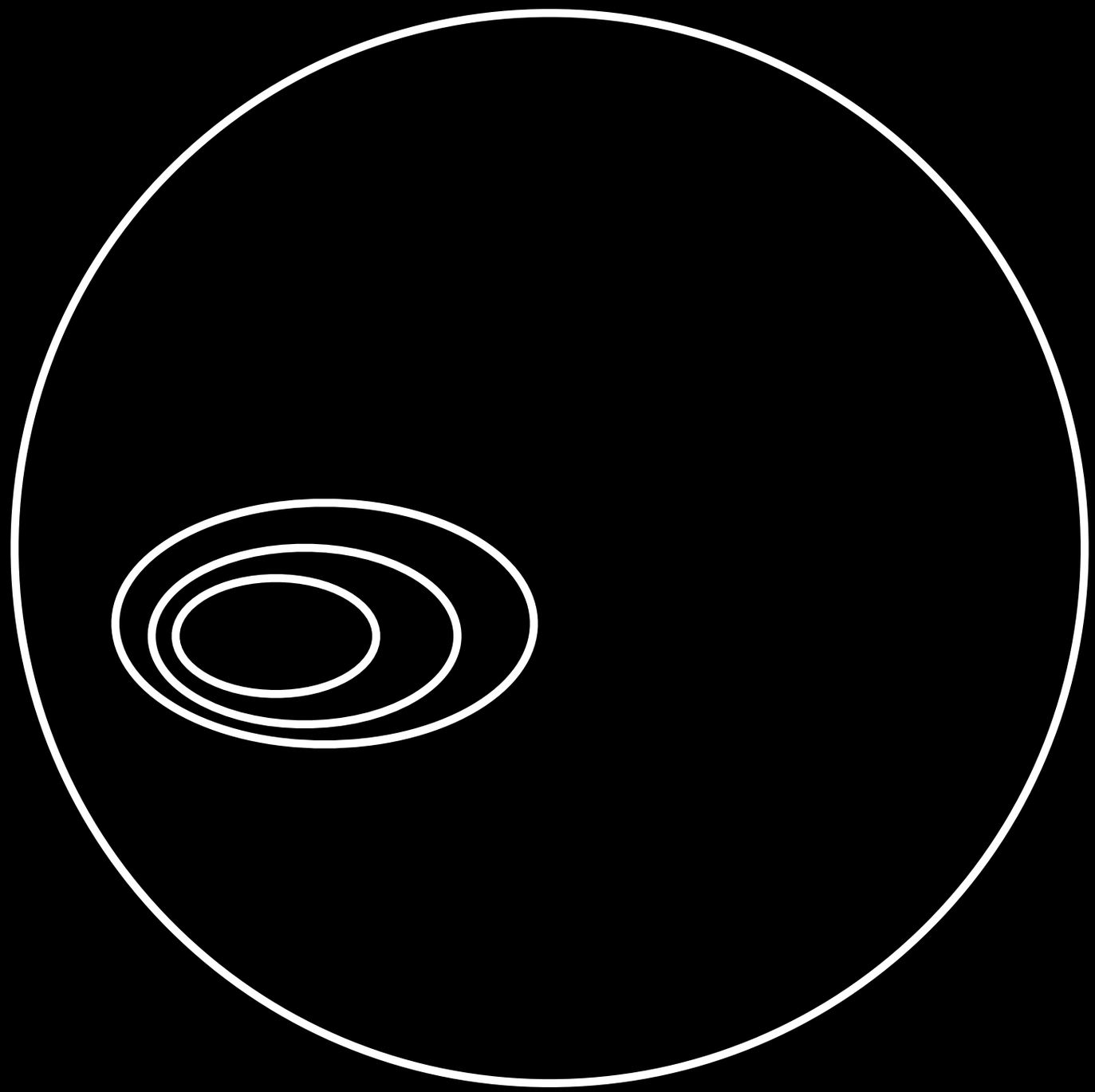
# UNDER-APPROXIMATE VERIFICATION

- Parametrised



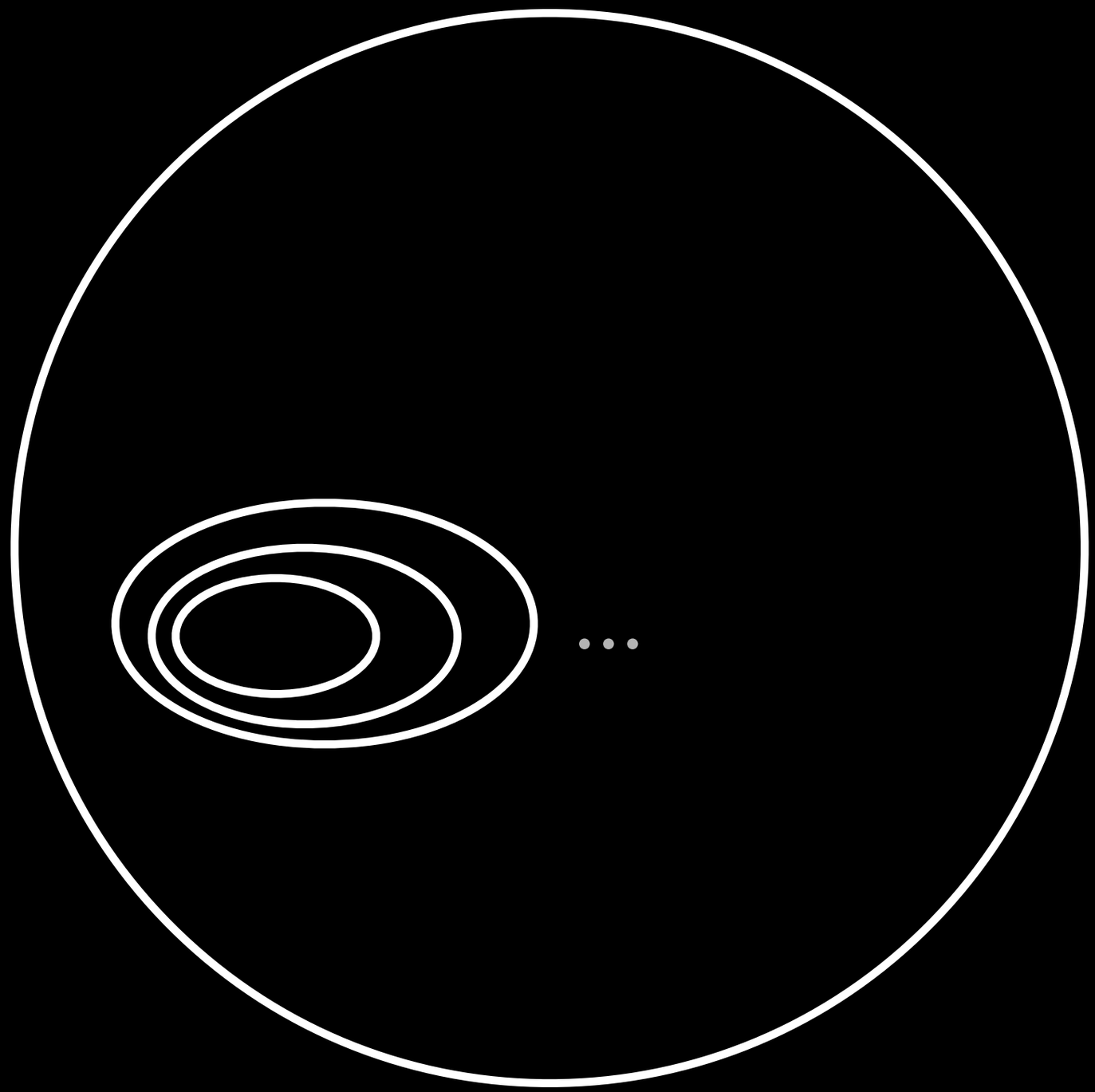
# UNDER-APPROXIMATE VERIFICATION

- Parametrised



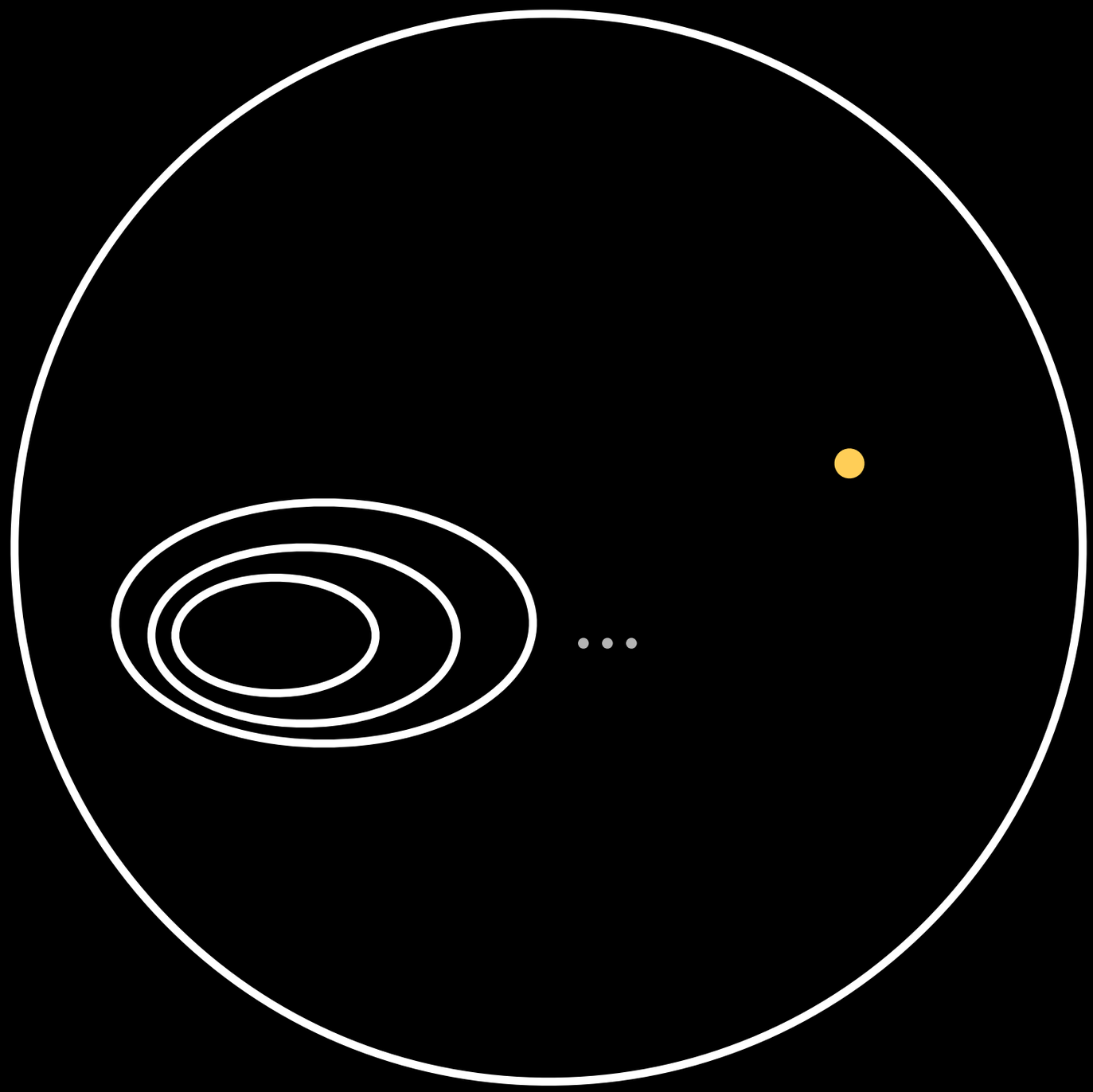
# UNDER-APPROXIMATE VERIFICATION

- Parametrised



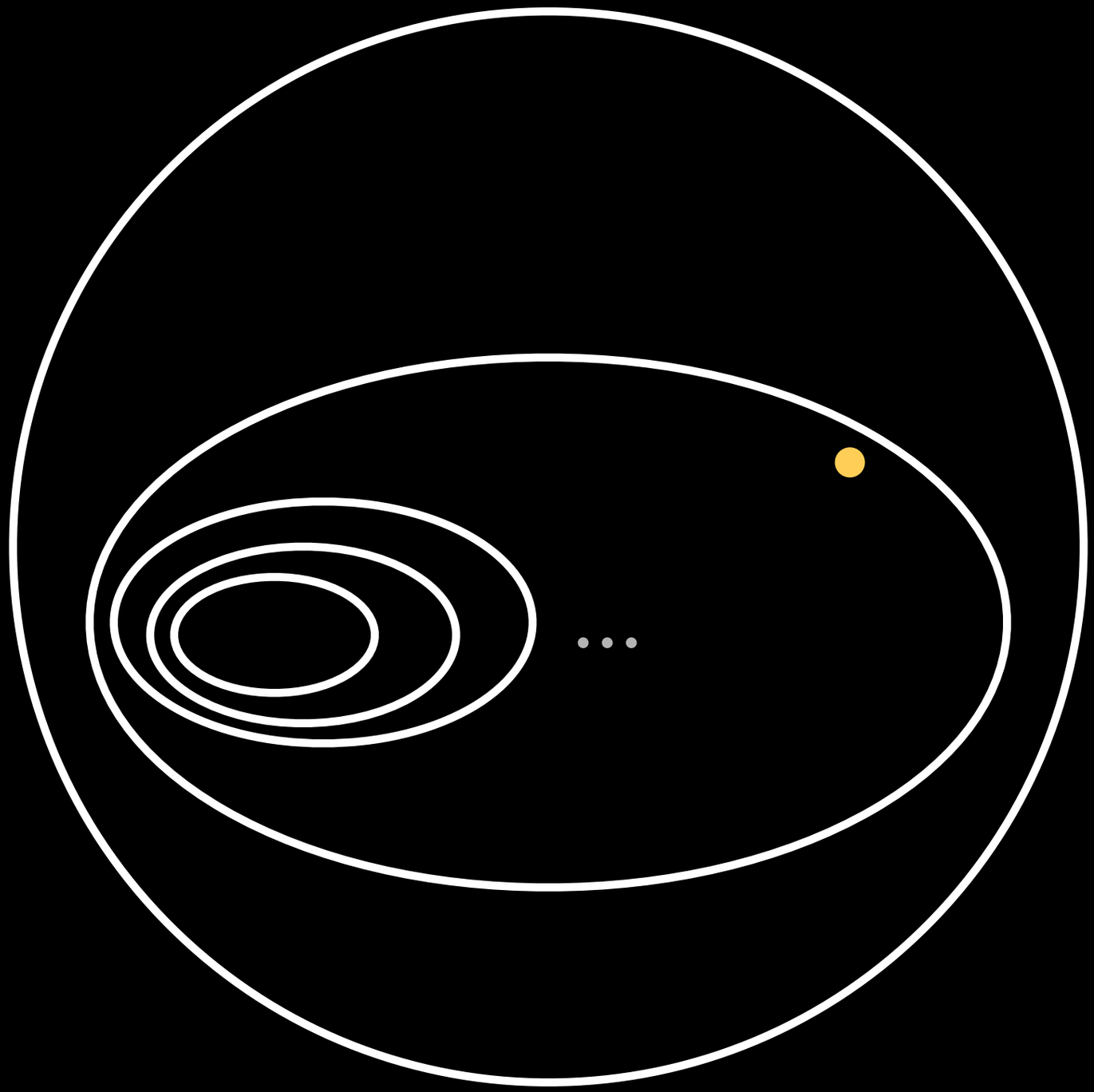
# UNDER-APPROXIMATE VERIFICATION

- Parametrised
- Exhaustive



# UNDER-APPROXIMATE VERIFICATION

- Parametrised
- Exhaustive

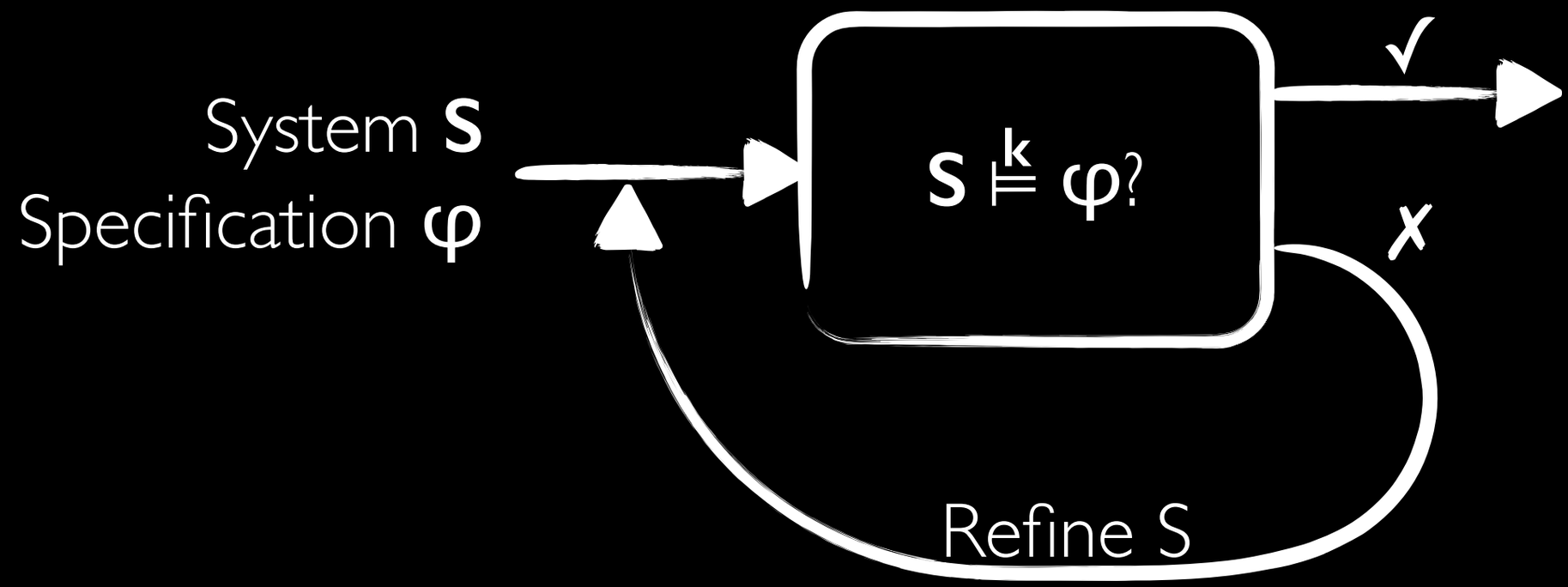


# UNDER-APPROXIMATE VERIFICATION

**Model Checking**

**>**

**Decidable**

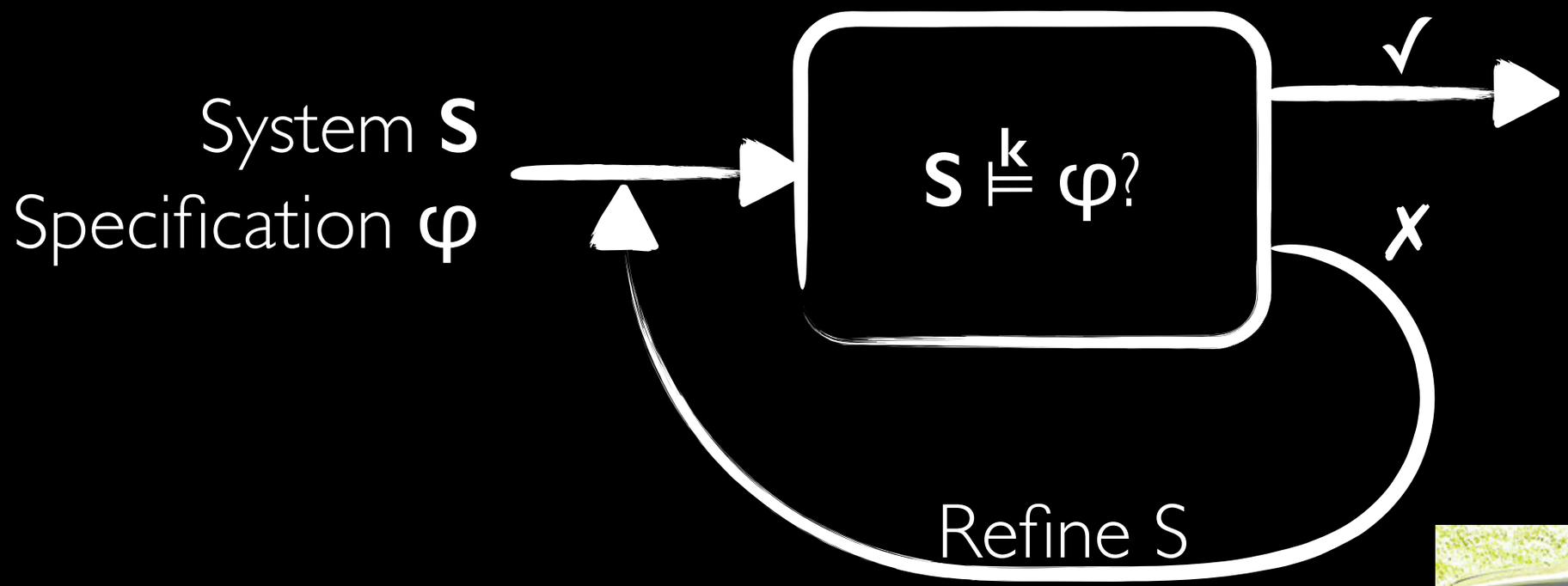


# UNDER-APPROXIMATE VERIFICATION

**Model Checking**

**>**

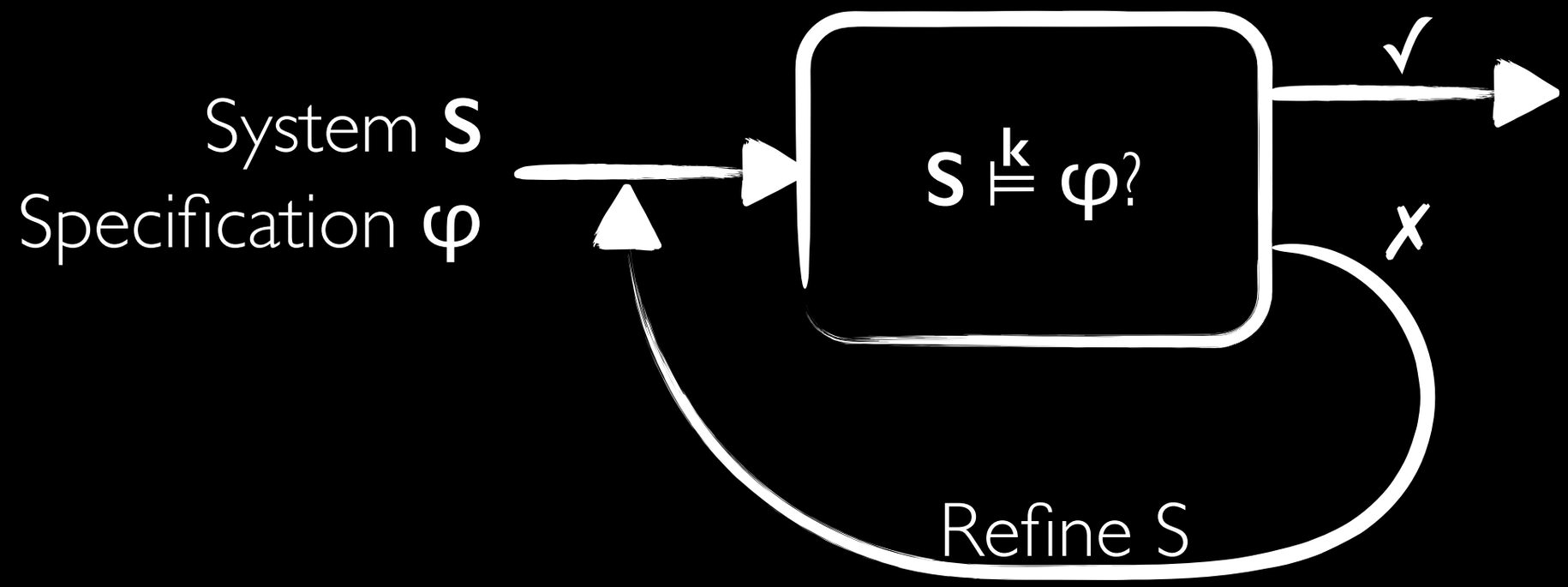
**Decidable**

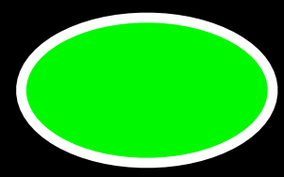



# UNDER-APPROXIMATE VERIFICATION

**Model Checking**

**>  Decidable**

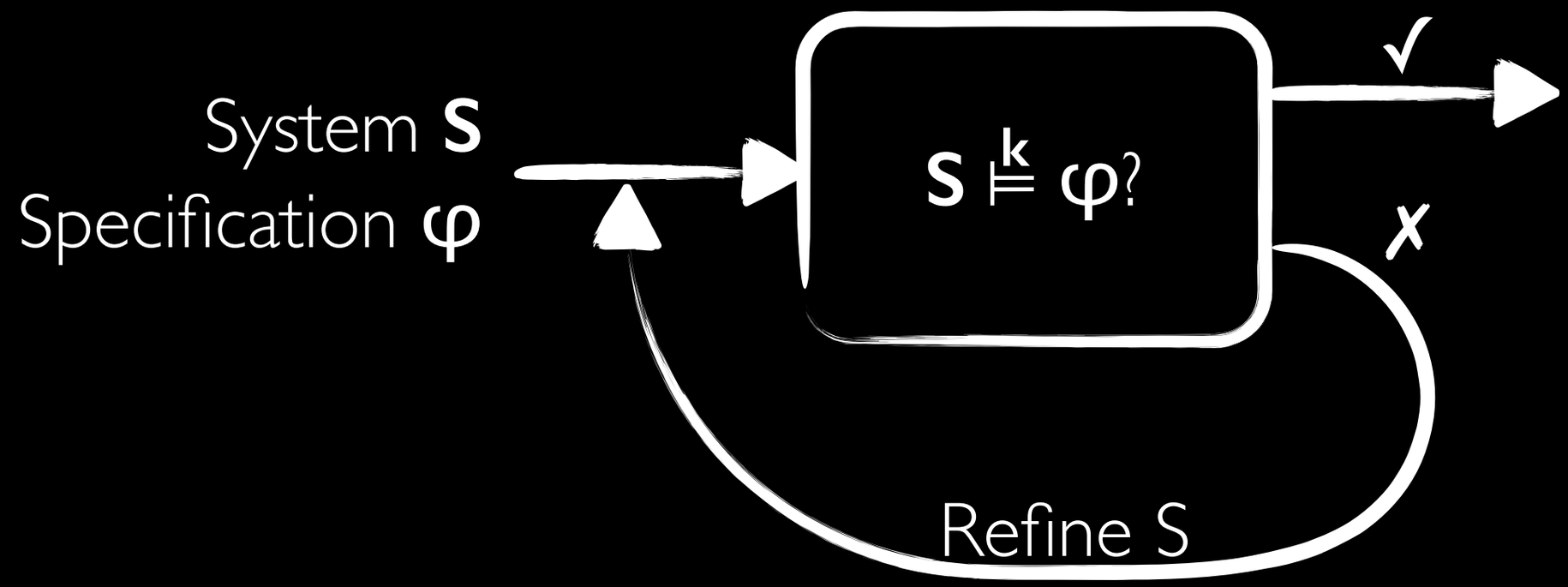


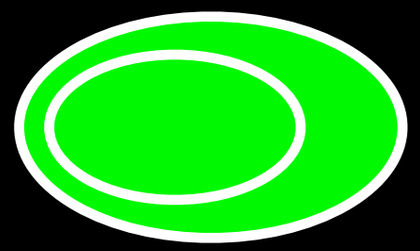



# UNDER-APPROXIMATE VERIFICATION

**Model Checking**

**>  Decidable**



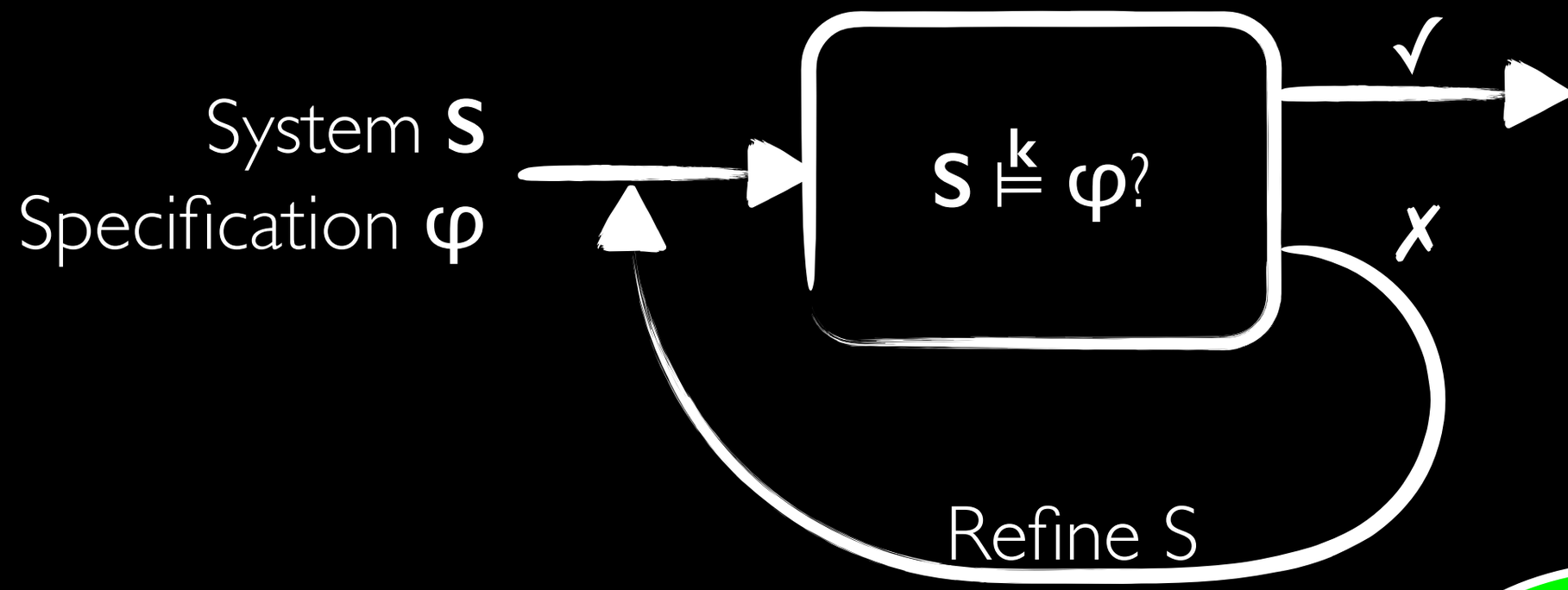



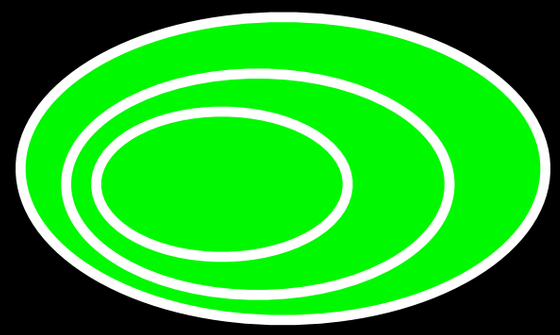
# UNDER-APPROXIMATE VERIFICATION

**Model Checking**

**>**

**Decidable**

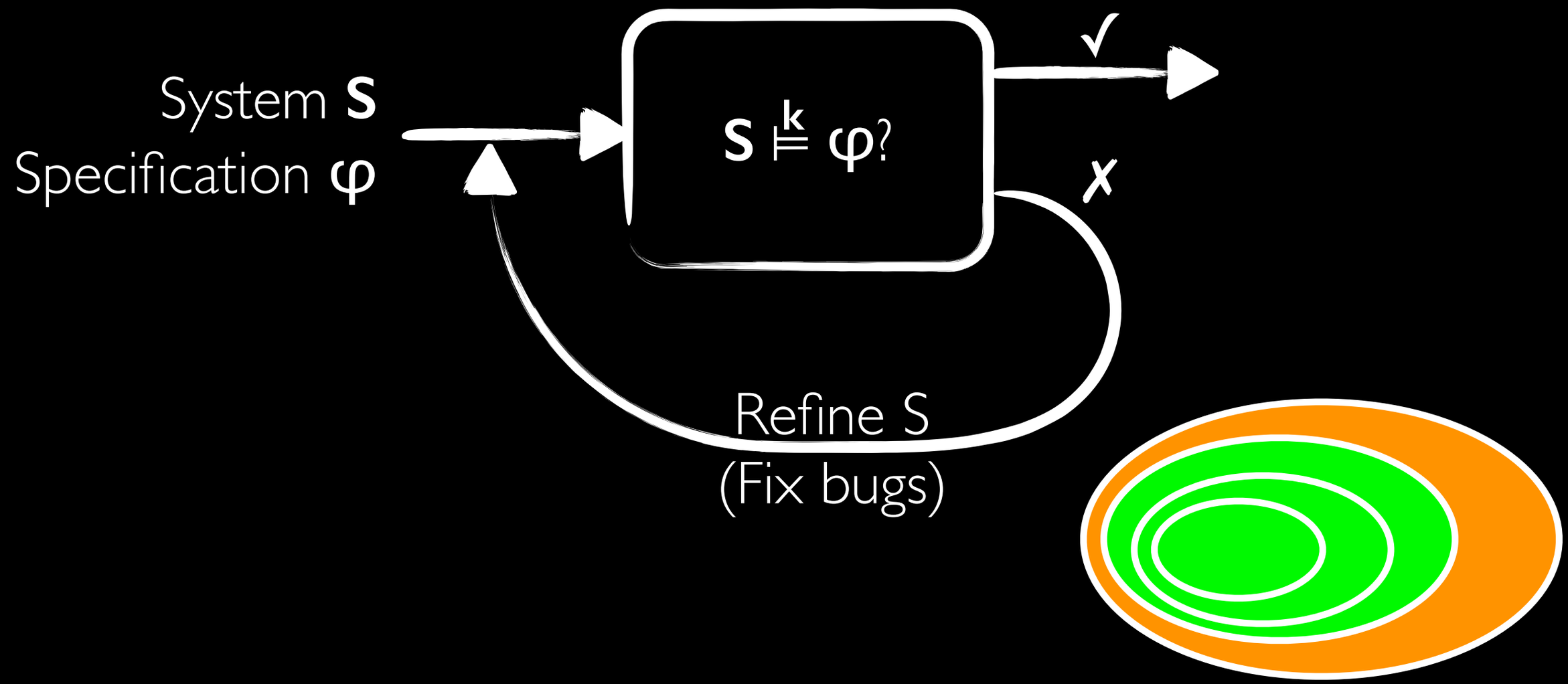


# UNDER-APPROXIMATE VERIFICATION

**Model Checking**

**>  Decidable**

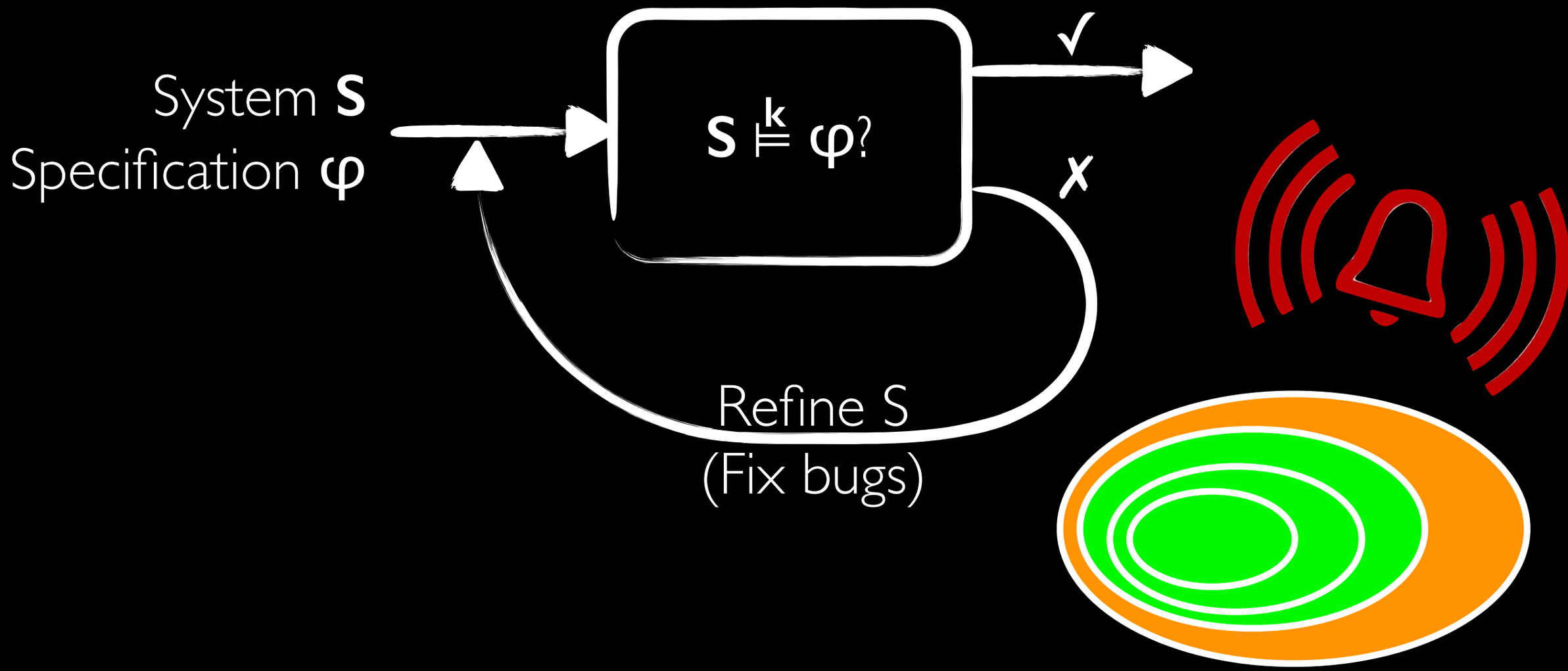


# UNDER-APPROXIMATE VERIFICATION

**Model Checking**

**>**

**Decidable**

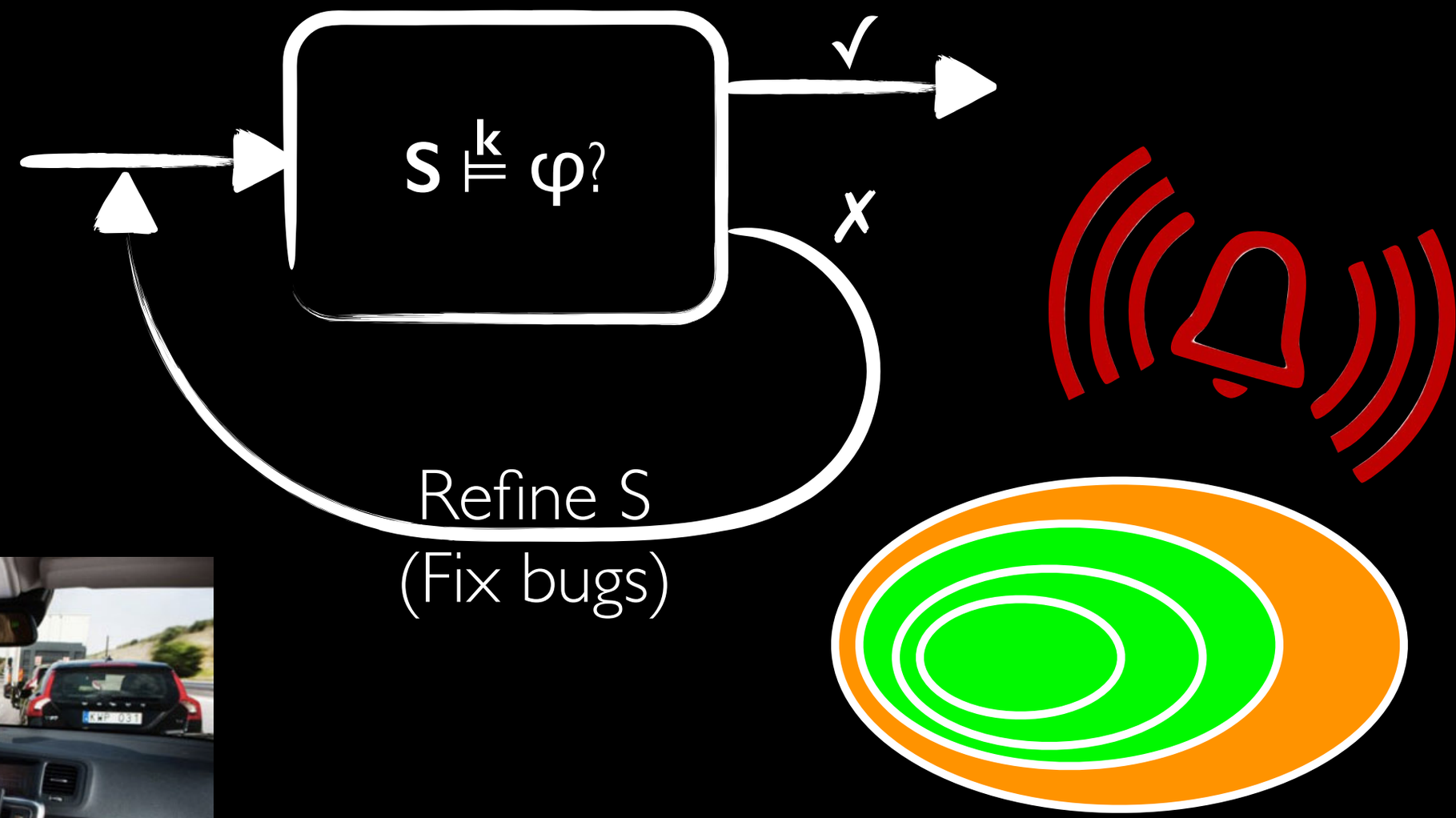


# UNDER-APPROXIMATE VERIFICATION

**Model Checking**

**>**

**Decidable**



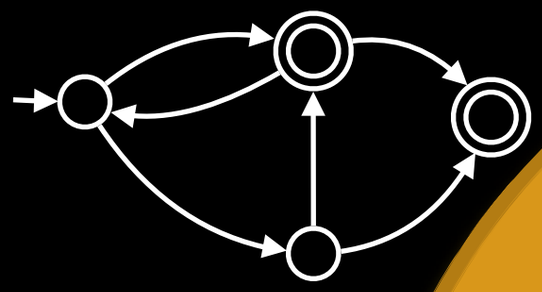
# CONTROLLERS FOR VERIFICATION OF DISTRIBUTED SYSTEMS

# COMMUNICATING DISTRIBUTED SYSTEMS

Process 1

Network

Process 2



Process 4

Process 3

# COMMUNICATING DISTRIBUTED SYSTEMS

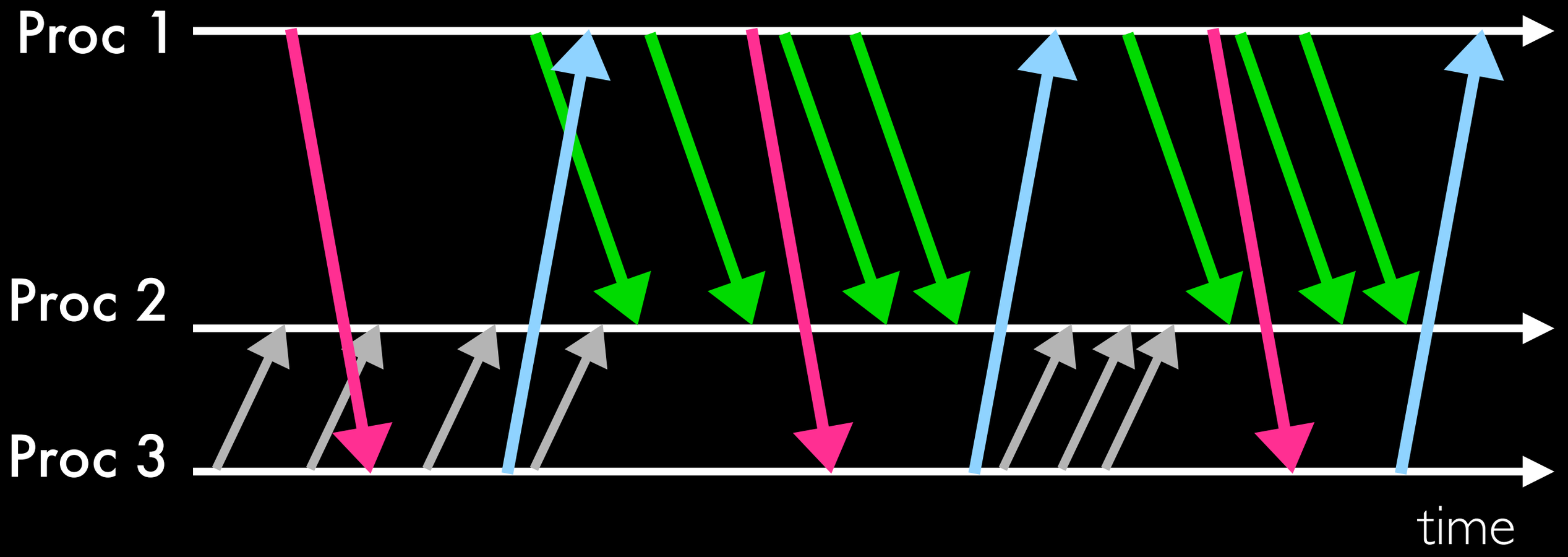

Process 1

Process 2

Process 3

Network

# BEHAVIOURS : MESSAGE SEQUENCE CHARTS



# CONTROLLERS FOR VERIFICATION OF DISTRIBUTED SYSTEMS

# COMMUNICATING DISTRIBUTED SYSTEMS


Process 1

Process 2

Process 3

Network

# CONTROLLERS FOR DISTRIBUTED SYSTEMS


Process 1

Process 2

Process 3

Controller 1

Controller 2

Controller 3

Network

# CONTROLLERS FOR DISTRIBUTED SYSTEMS


Process 1

Process 2

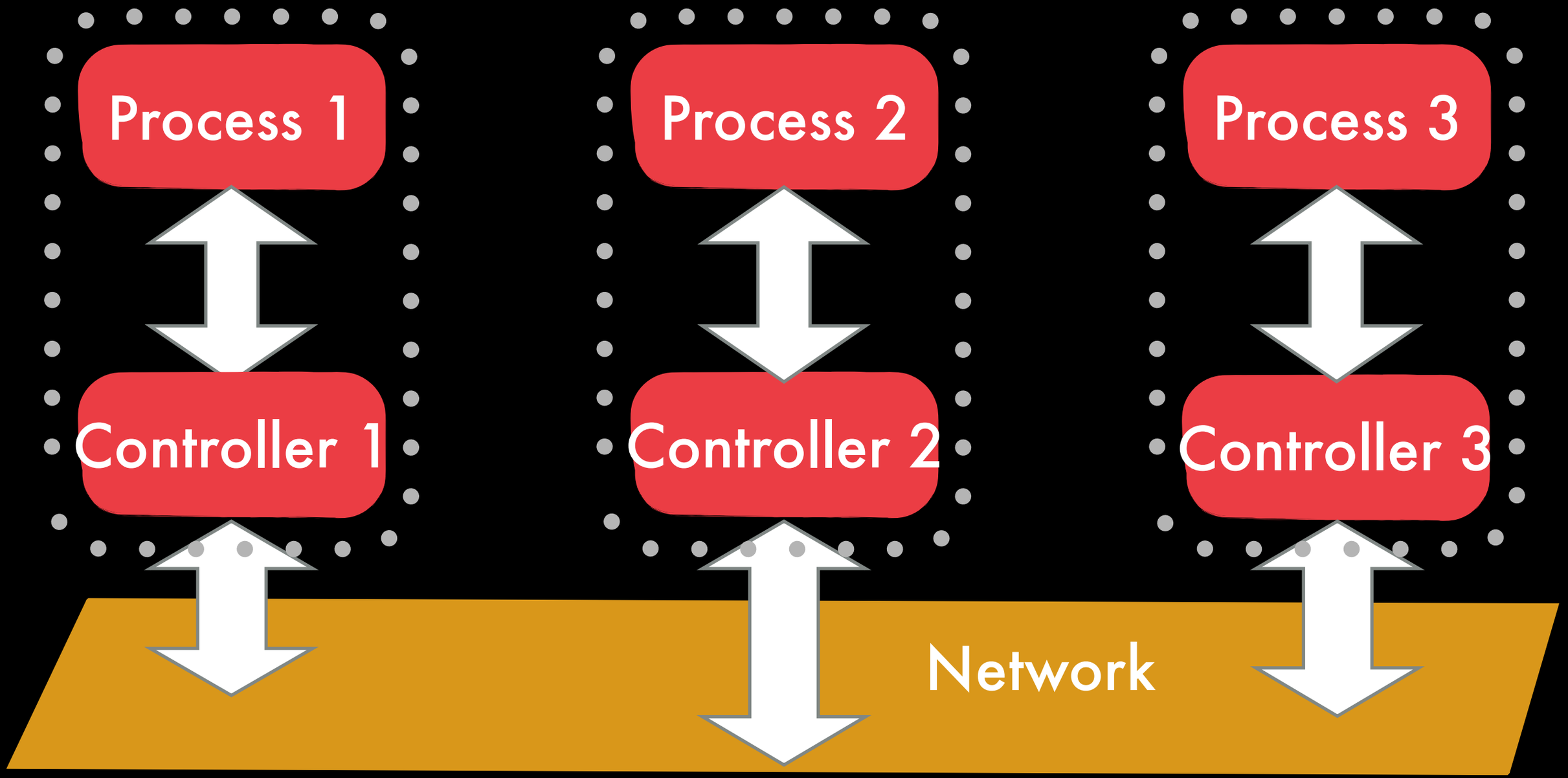
Process 3

Controller 1

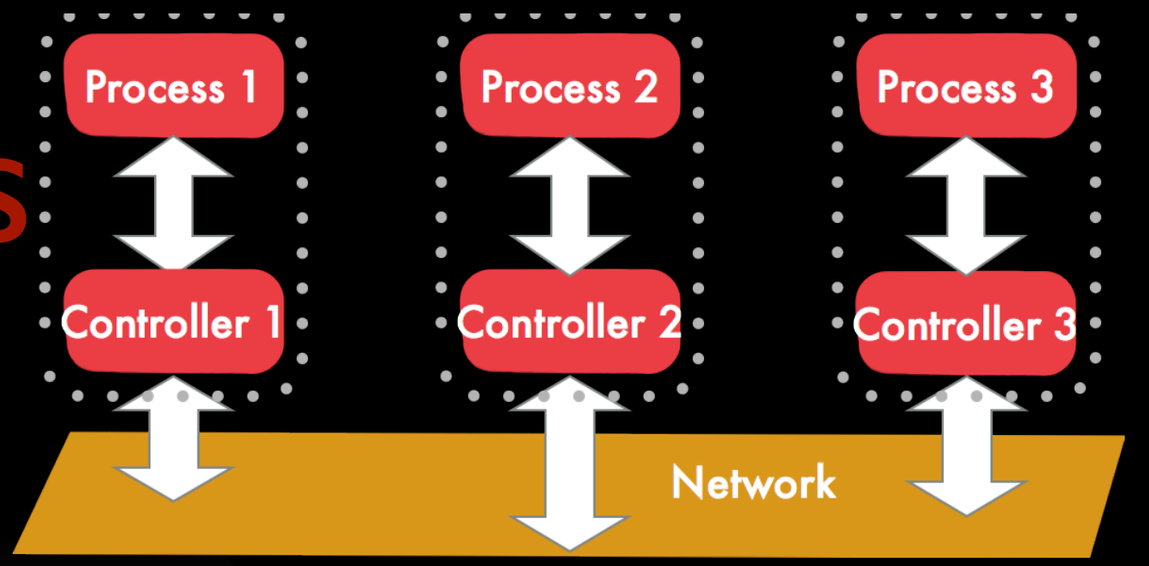
roller 2

Controller 3


Network

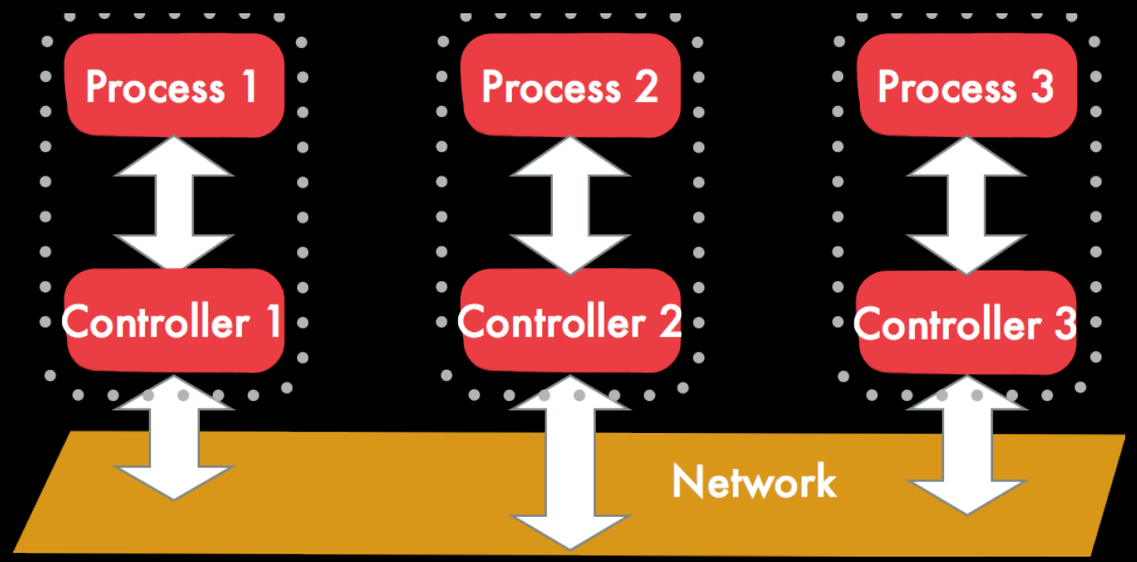

Process 1
Process 2
Process 3
Controller 1
Controller 2
Controller 3
Network

# CONTROLLERS FOR DISTRIBUTED SYSTEMS




- Collection of local controllers
- Communication via piggy-backing
- Privacy: Do NOT read states/messages

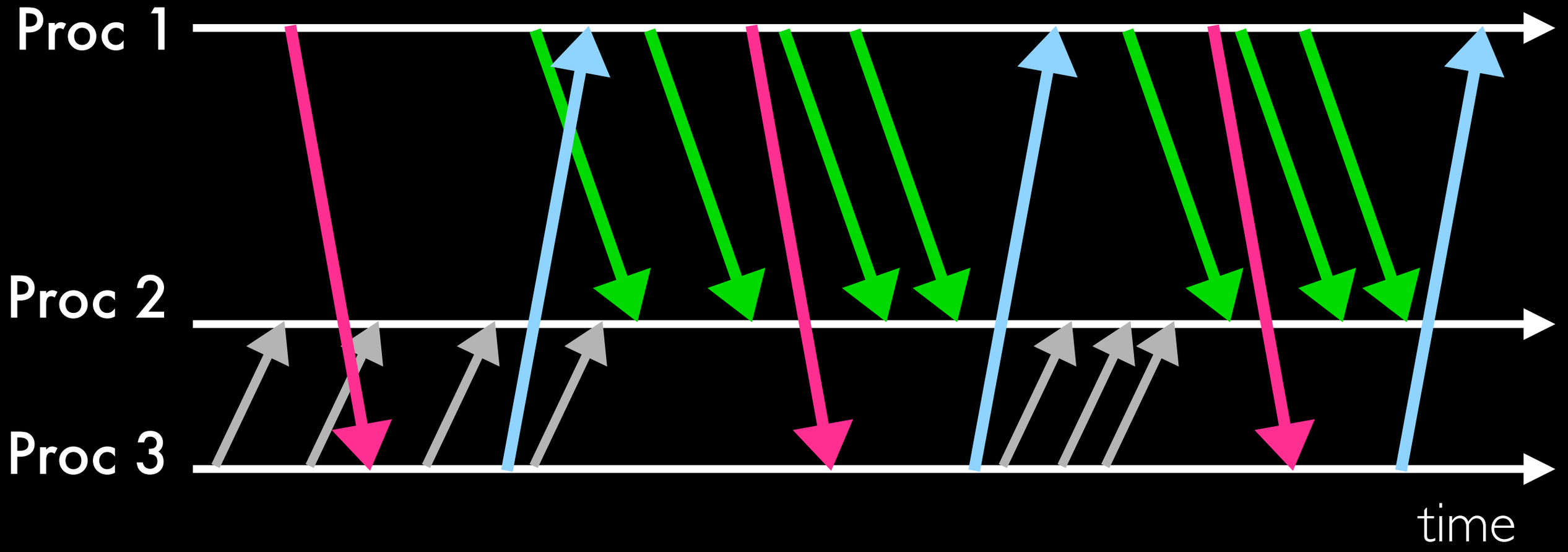
# LET'S DESIGN A CONTROLLER



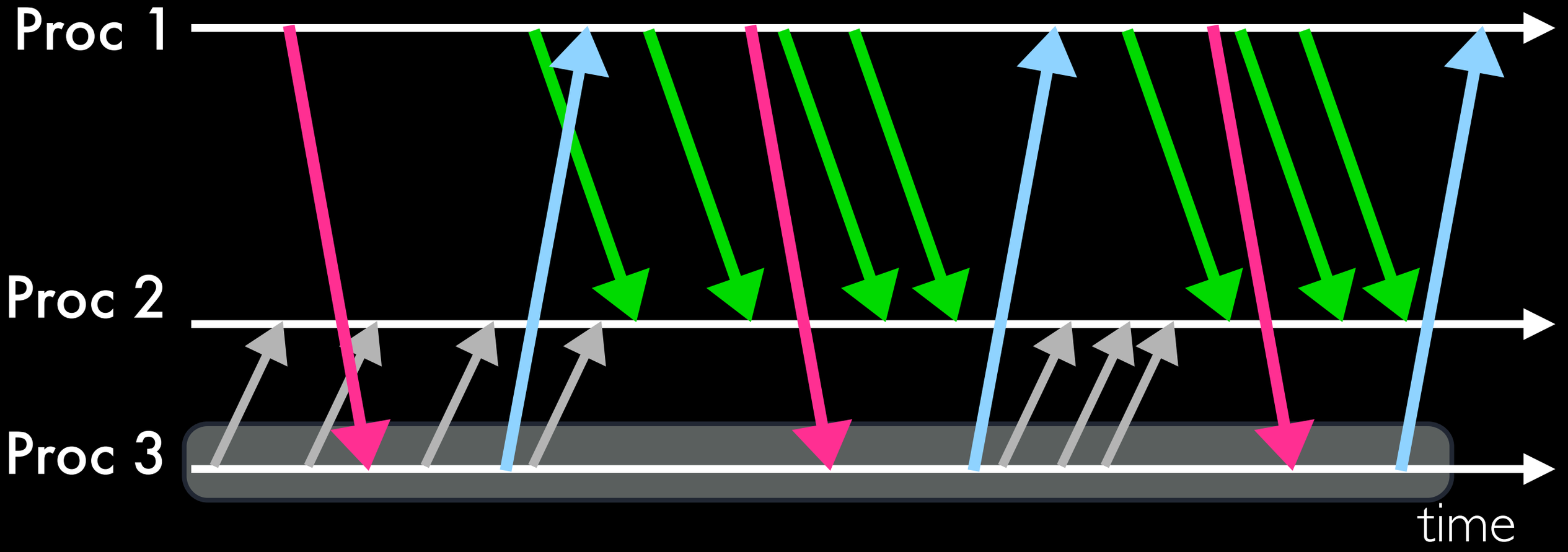

## UNDER-APPROXIMATION: BOUNDED (*K*) PHASE

- Bounded number (k) of phases
- Phase: Receive from one process, send to all processes
- No cycles

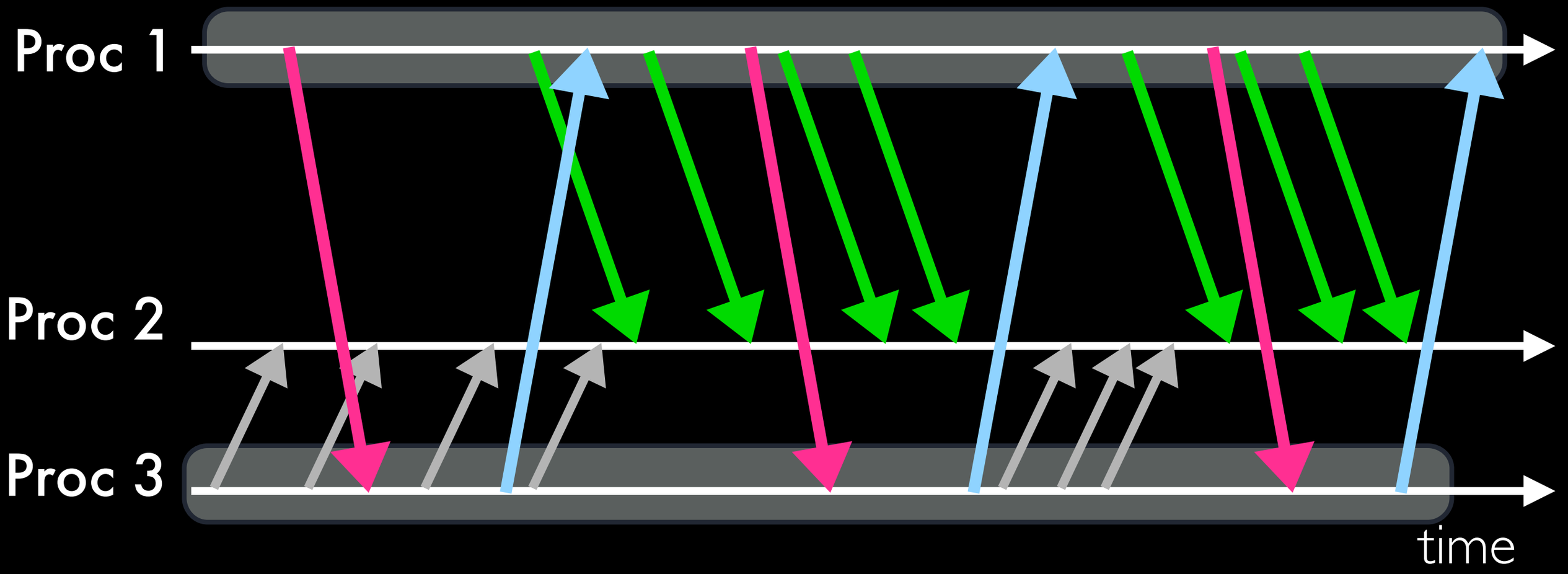
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



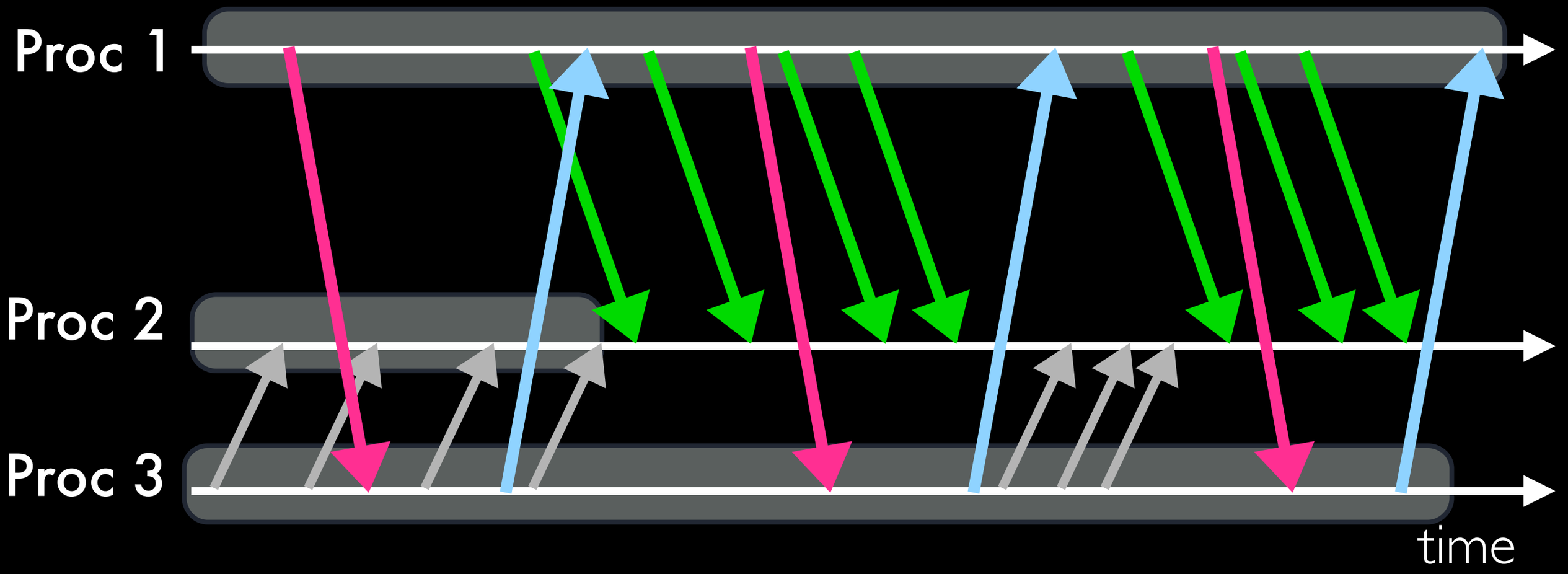
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



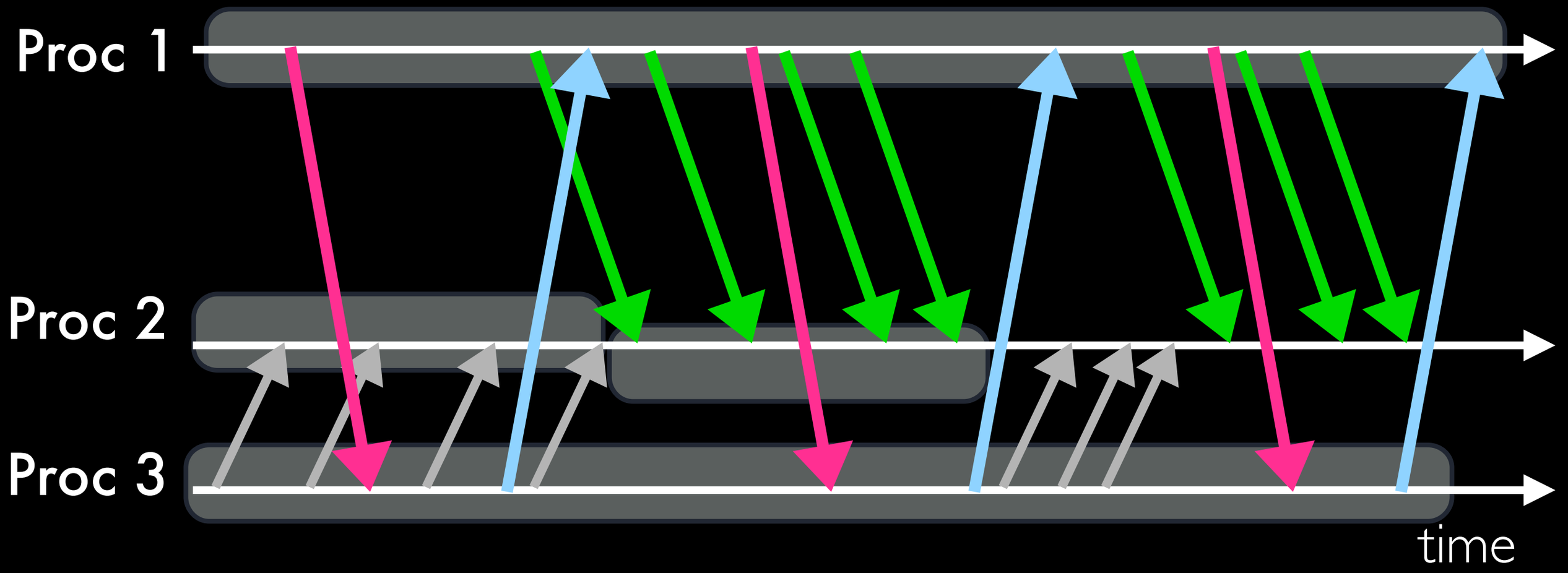
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



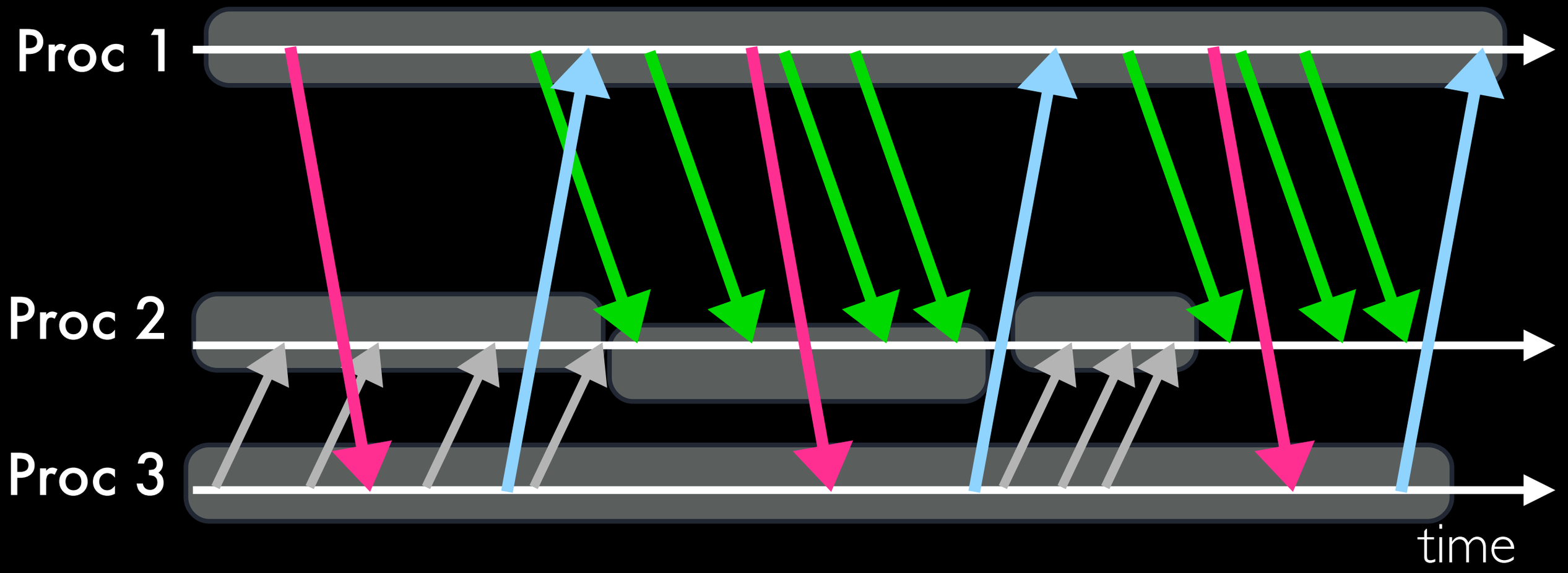
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



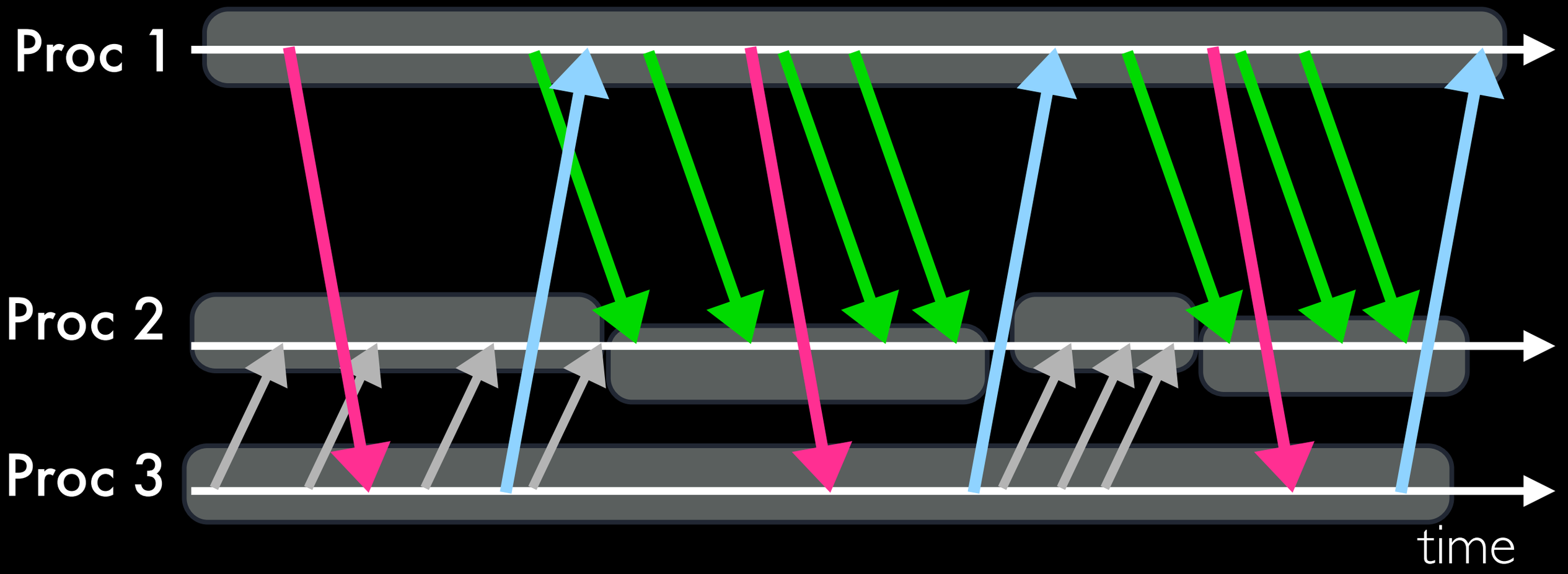
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



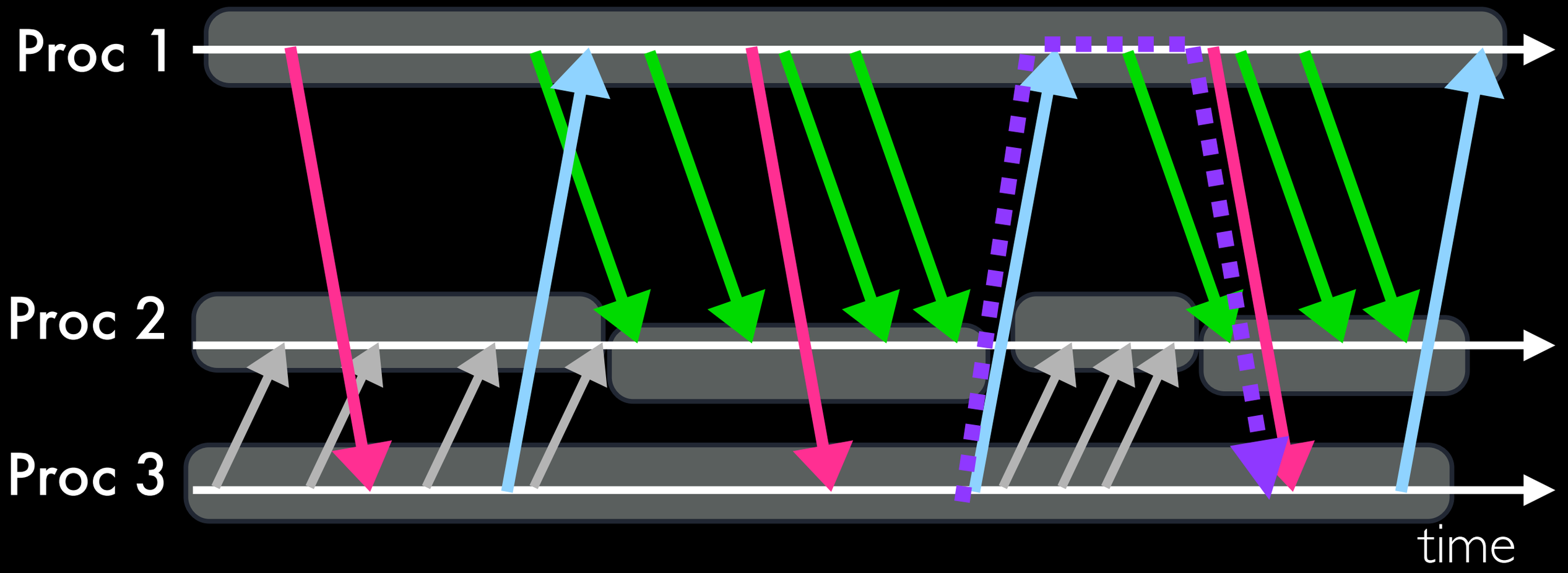
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



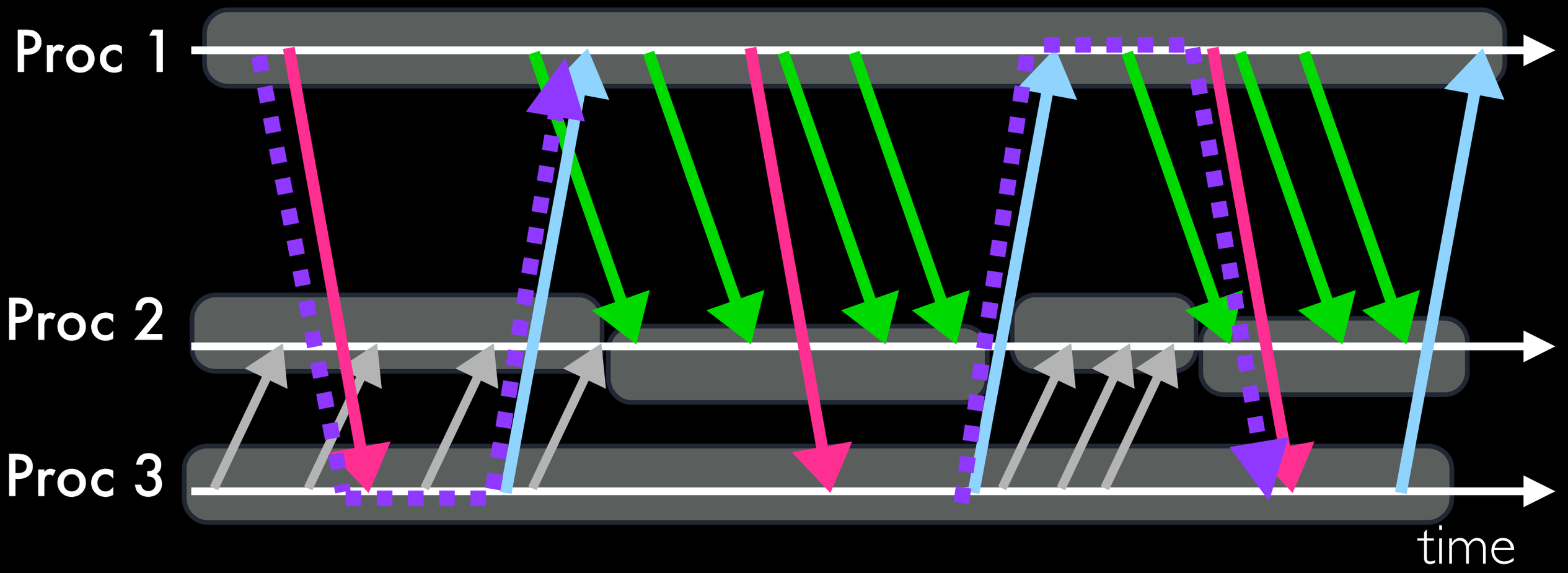
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



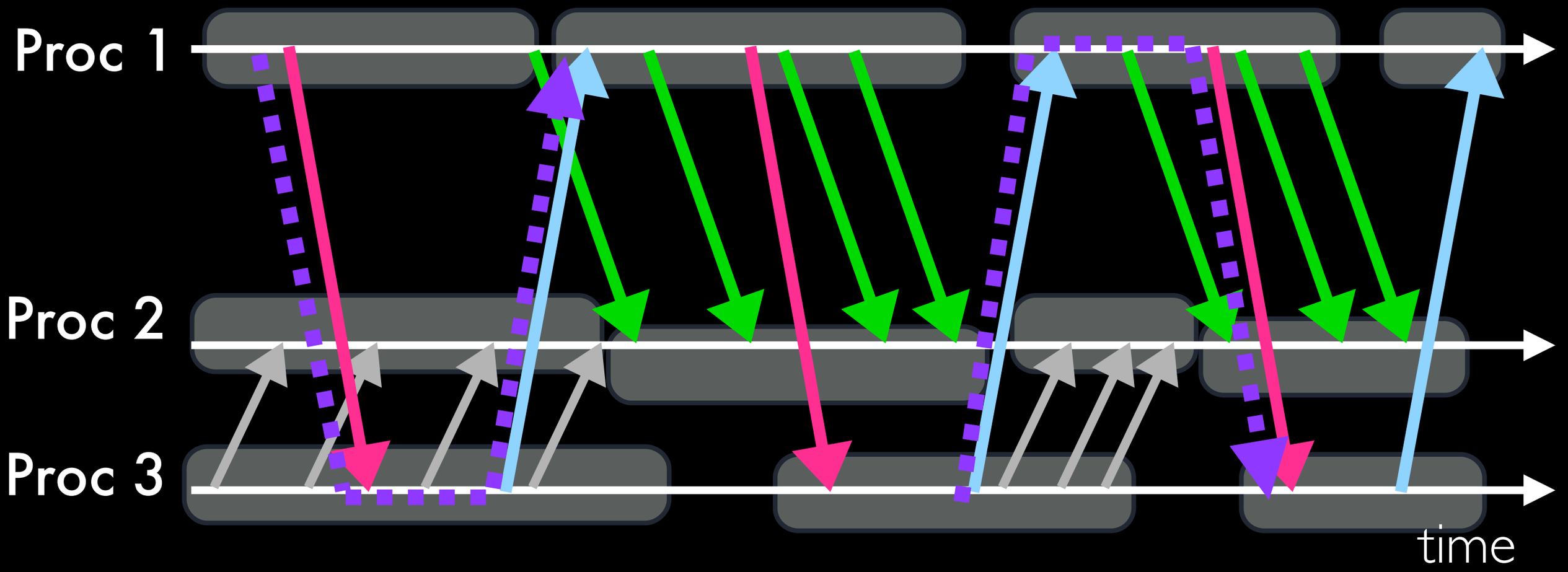
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



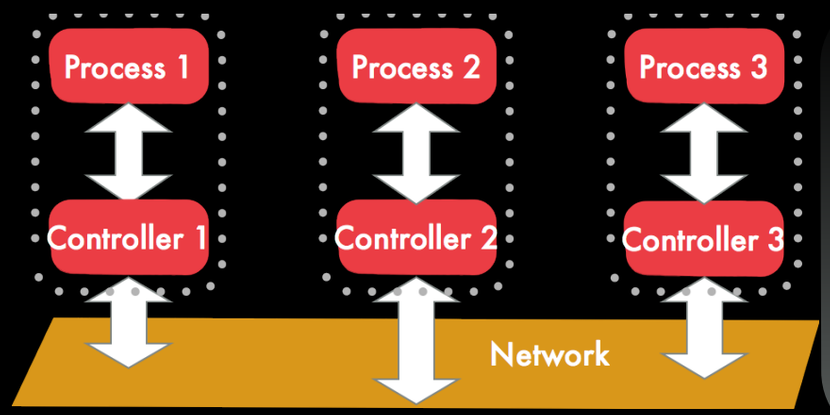
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



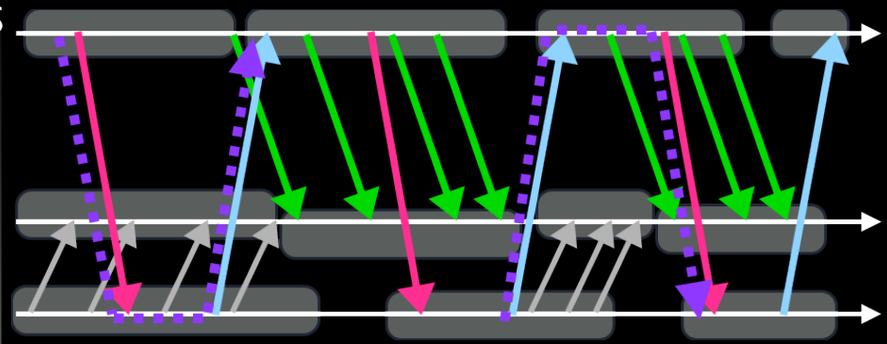
- Bounded number (k) of phases
- Phase: Receive from one proc, send to all procs
- No cycles



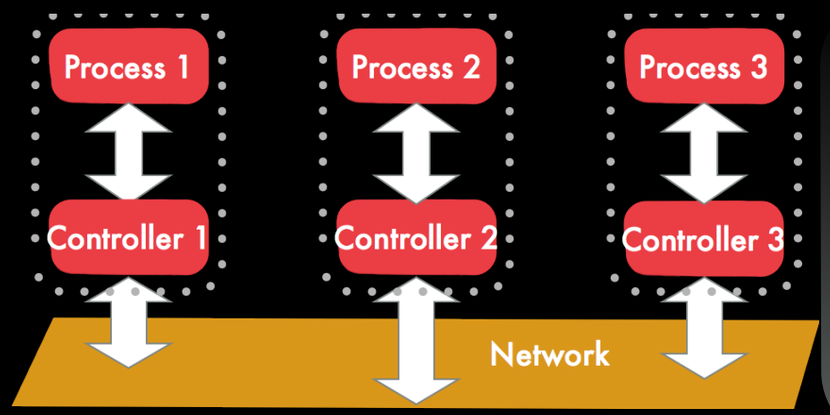
# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A



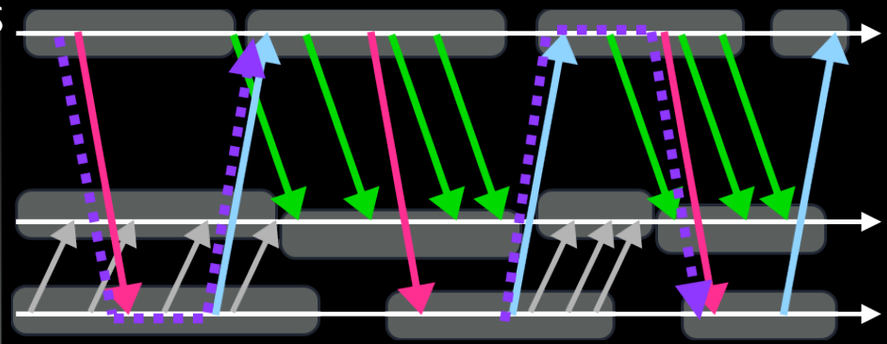

- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles



# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

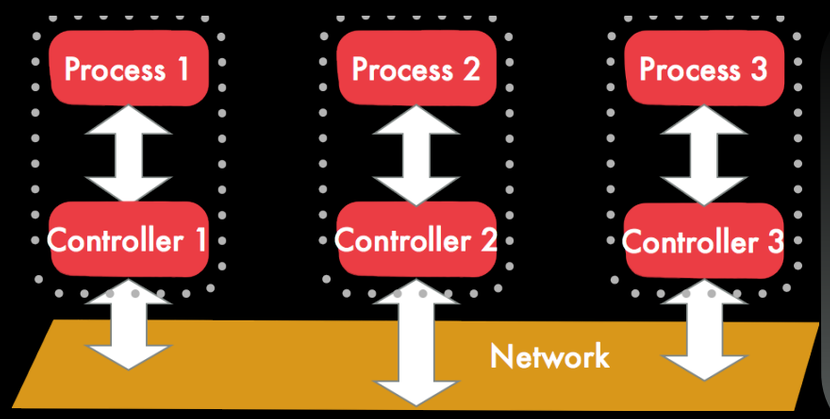



- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles

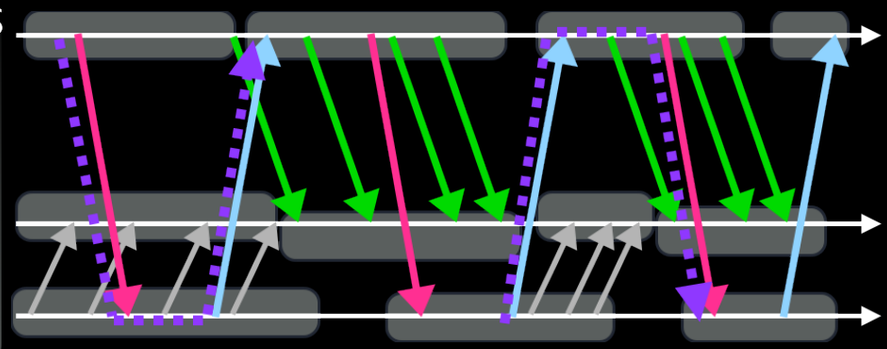


A local controller for each process

# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A




- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles

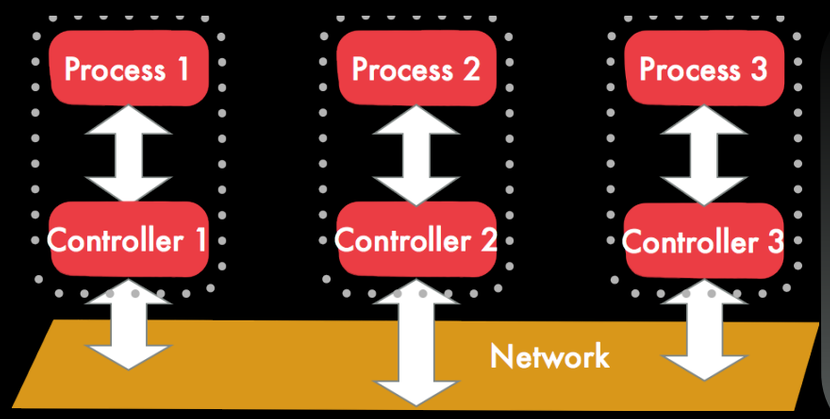


A local controller for each process

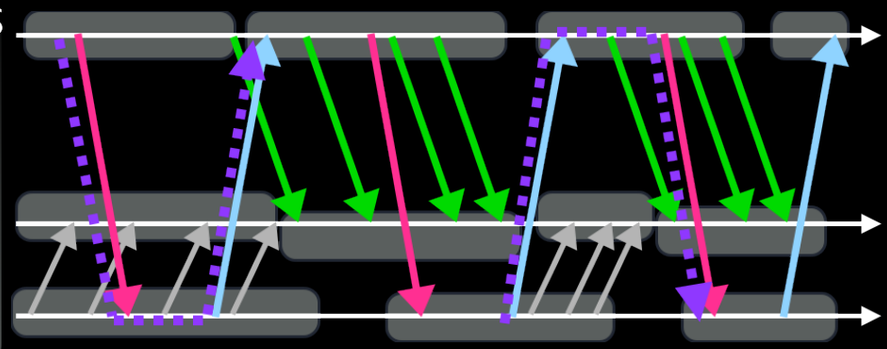
Has a Phase Counter

Remembers current sender

# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A




- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles



A local controller for each process

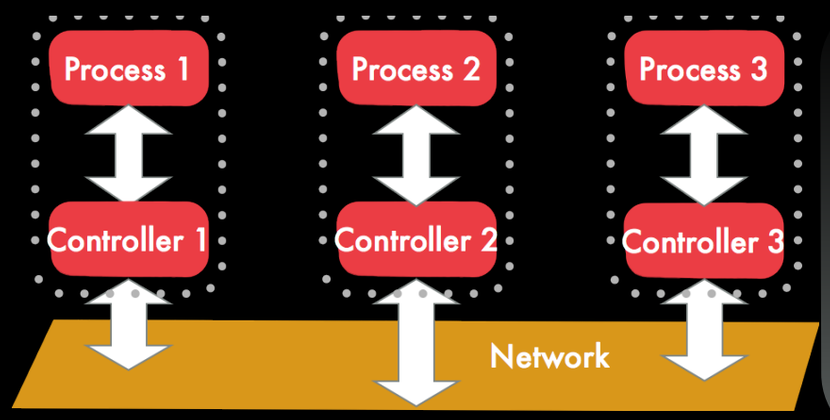
Has a Phase Counter

Remembers current sender

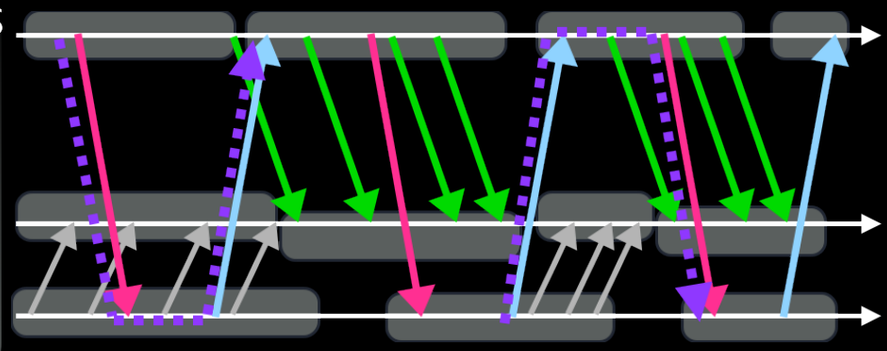
Different sender?

Detect Cycle?

# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A




- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles



A local controller for each process

Has a Phase Counter

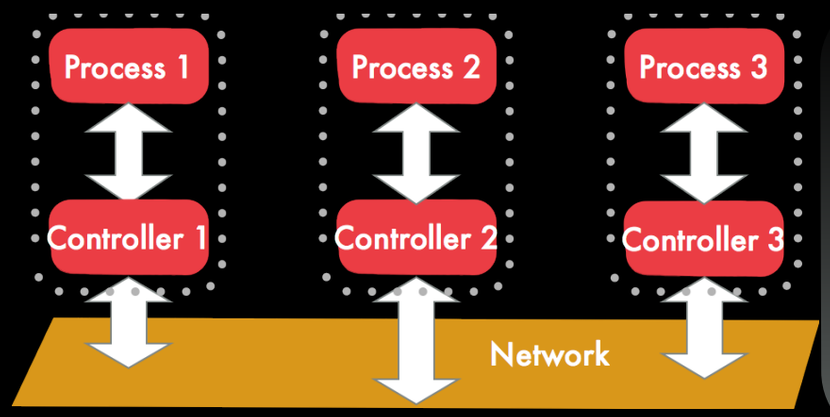
Remembers current sender

Different sender?

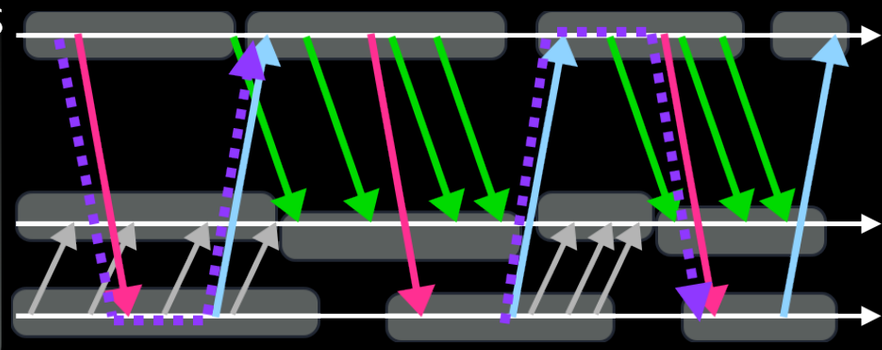
Detect Cycle?

Increment counter,
Update channel

# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A




- Bounded number (k) of phases
- Phase: Receive from one proc, send to all processes
- No cycles



A local controller for each process

State

- Has a Phase Counter
- Remembers current sender

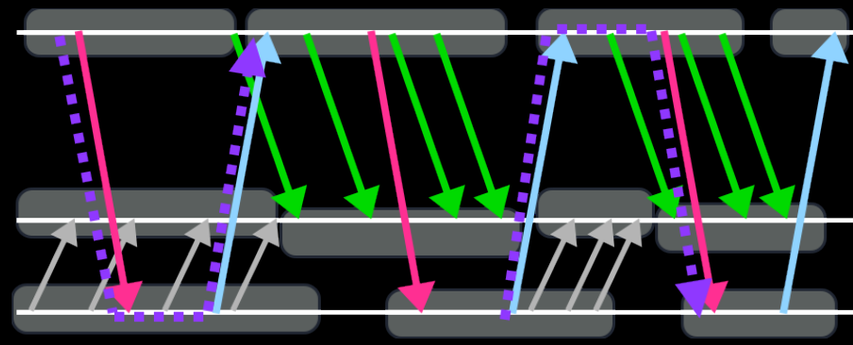
Transitions

- Different sender?
- Detect Cycle?

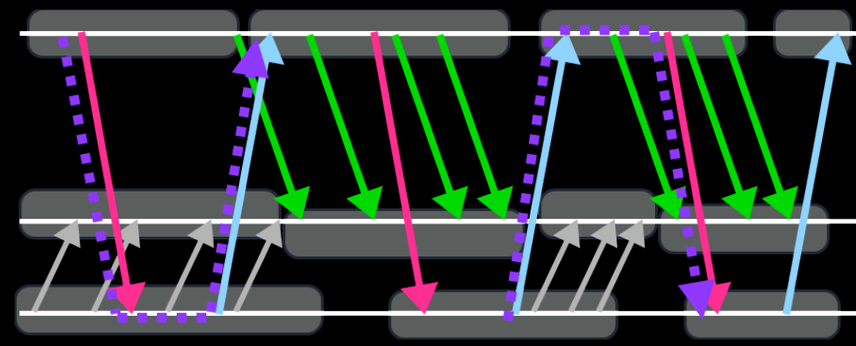
Increment counter,
Update channel

# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

Detect Cycle?



# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

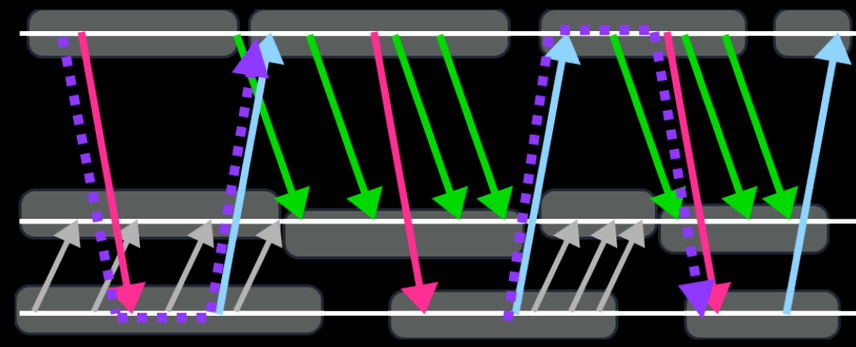


Detect Cycle?

Phase Vectors

# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A

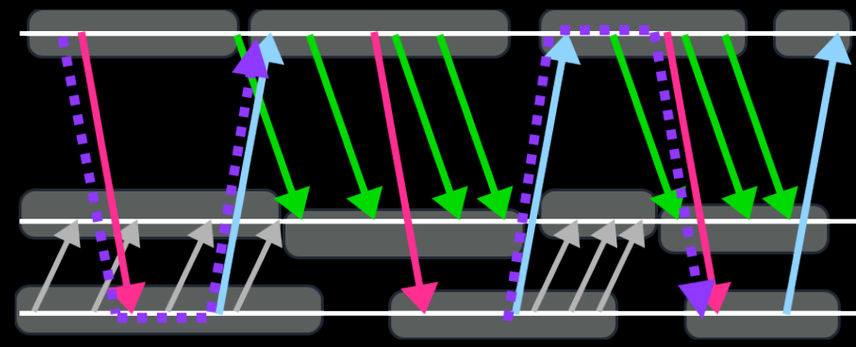
Detect Cycle?



Phase Vectors

best info about phase number of other processes

# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A



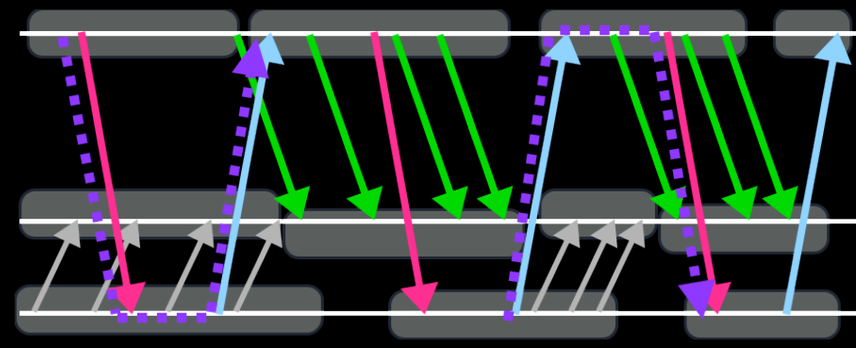
Detect Cycle?

Phase Vectors

best info about phase number of other processes

Sends: tag with phase vector

# DISTRIBUTED CONTROLLER FOR K-BOUNDED PHASE U-A



Detect Cycle?

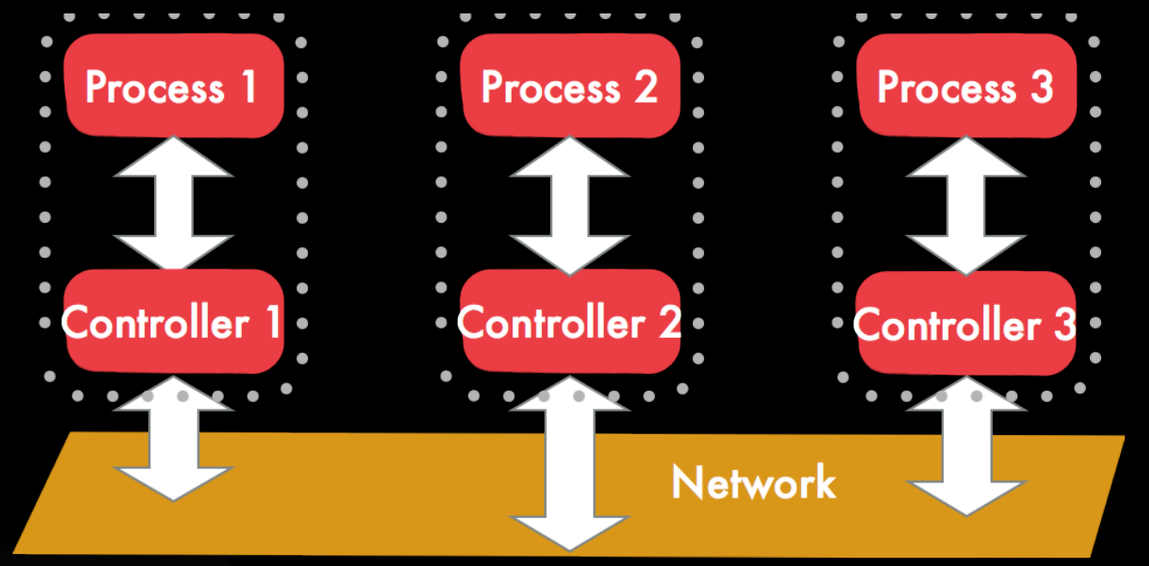
Phase Vectors

best info about phase number of other processes

Sends: tag with phase vector

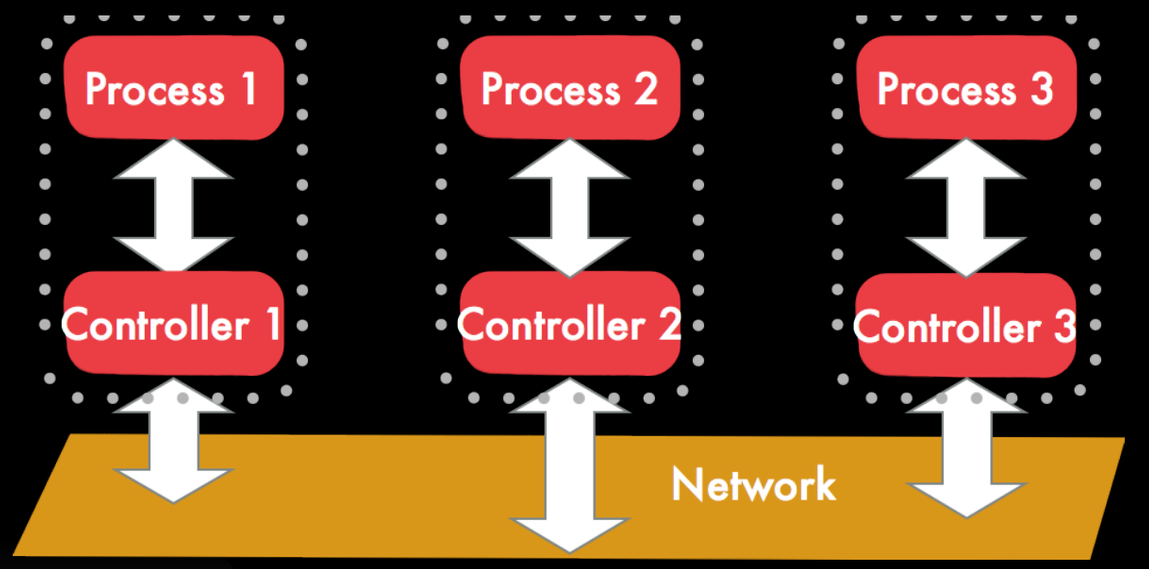
Receives: update phase vector by taking MAX

# CONTROLLERS FOR BOUNDED PHASE DISTRIBUTED SYSTEMS




- Collection of local controllers
- Communication via piggy-backing
- Privacy: Do NOT read states/messages

# CONTROLLERS FOR BOUNDED PHASE DISTRIBUTED SYSTEMS




- Collection of local controllers
- Communication via piggy-backing
- Privacy: Do NOT read states/messages

- System independent
- Generic
- Deterministic
- Finite state

# CONTROLLERS FOR VERIFICATION OF DISTRIBUTED SYSTEMS


- C. Aiswarya, Paul Gastin, and K. Narayan Kumar. Controllers for the verification of communicating multi-pushdown systems. In *CONCUR*'14, volume 8704 of *LNCS*, pages 297-311. Springer, 2014

- Aiswarya Cyriac. *Verification of Communicating Recursive Programs via Split-width*. PhD thesis, ENS Cachan, 2014.

Merci !