Kaapi & Dynaco

Remote Control of parallel run

~ 12 march 2008 ~

serge Guelton & jl Pazat

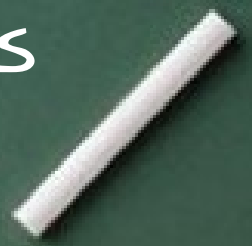
- IRISA / PARIS -

Safescale project



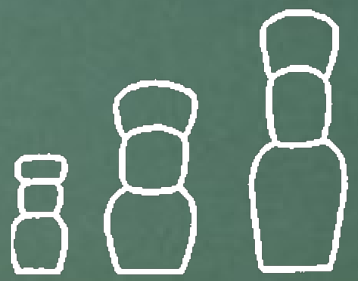
Plan

- ✓ Overview of Kaapi
- Overview of Dynaco
- Making them work together
- Results
- Conclusion & Future works

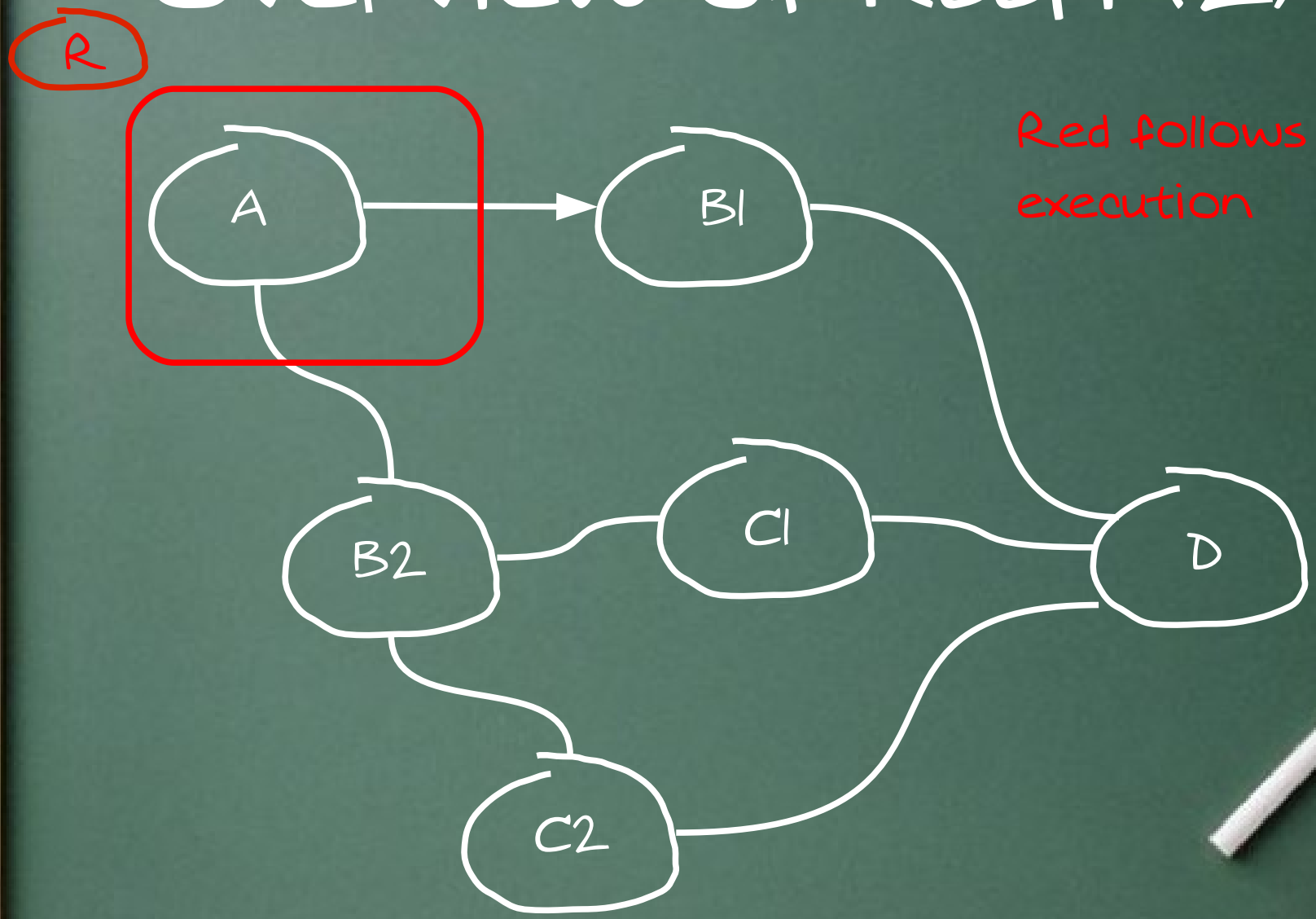


Overview of Kaapi (1/6)

- C++ library for distributed computing
- distributed work stealing (cf. CILK)
- dependencies analysis (cf. JADE)
- provides a high level API "Athapascan"
- developed in the MOAIS team (INRIA)



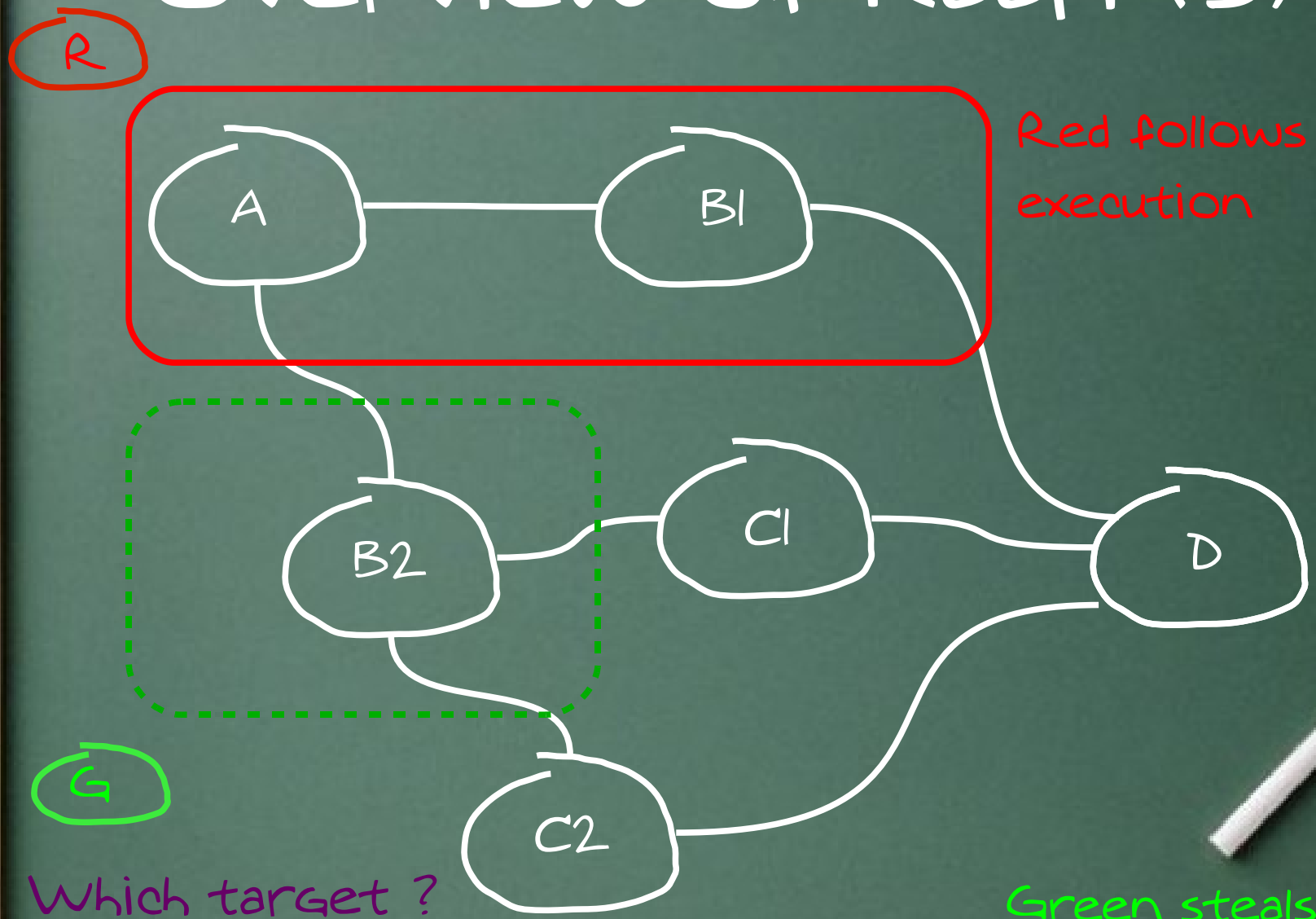
Overview of Kaapi (2/6)



Red follows sequential execution

Task flow of a dummy program

Overview of Kaapi (3/6)



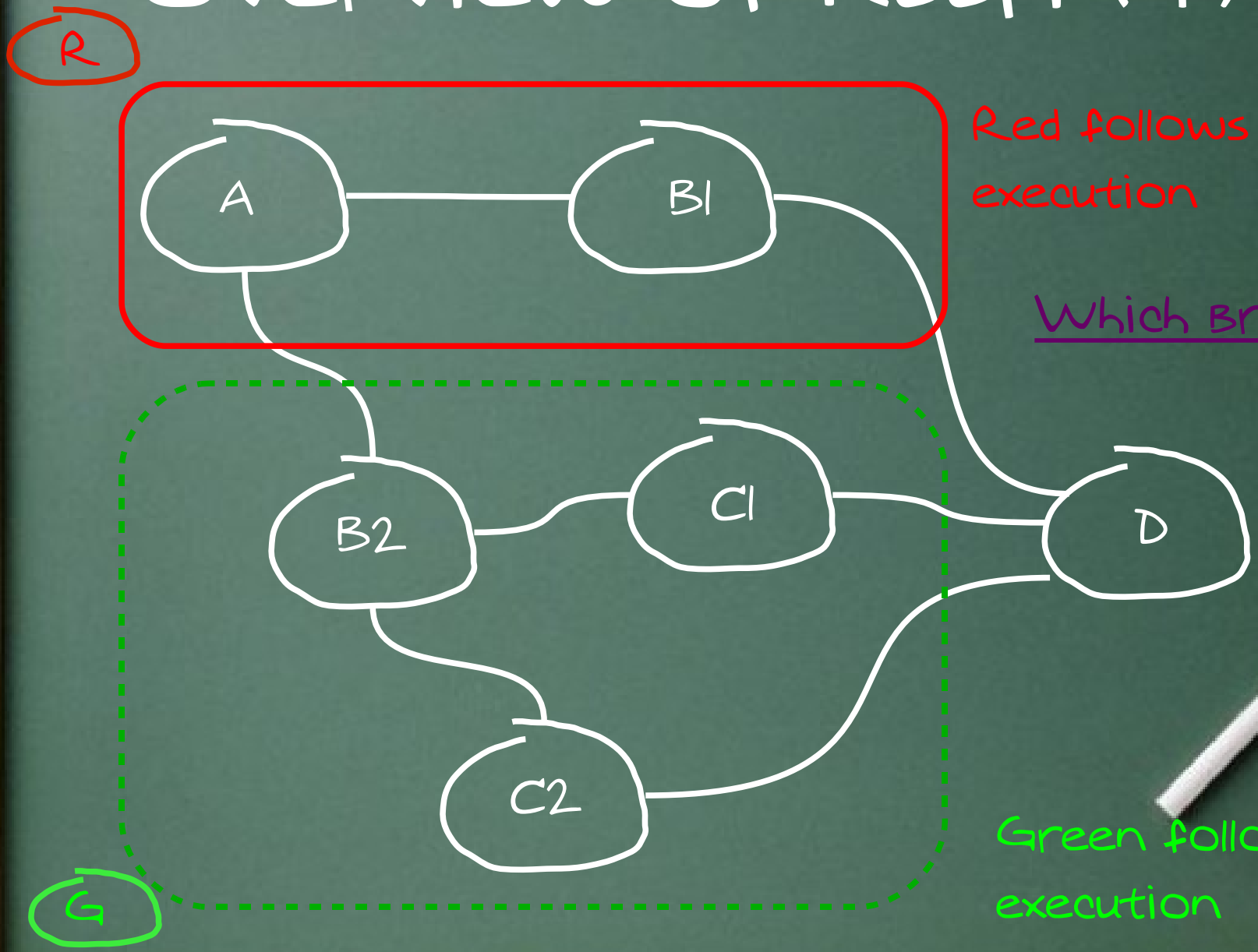
Red follows sequential execution

Green steals a task

Which target ?

Which stealer ?

Overview of Kaapi (4/6)

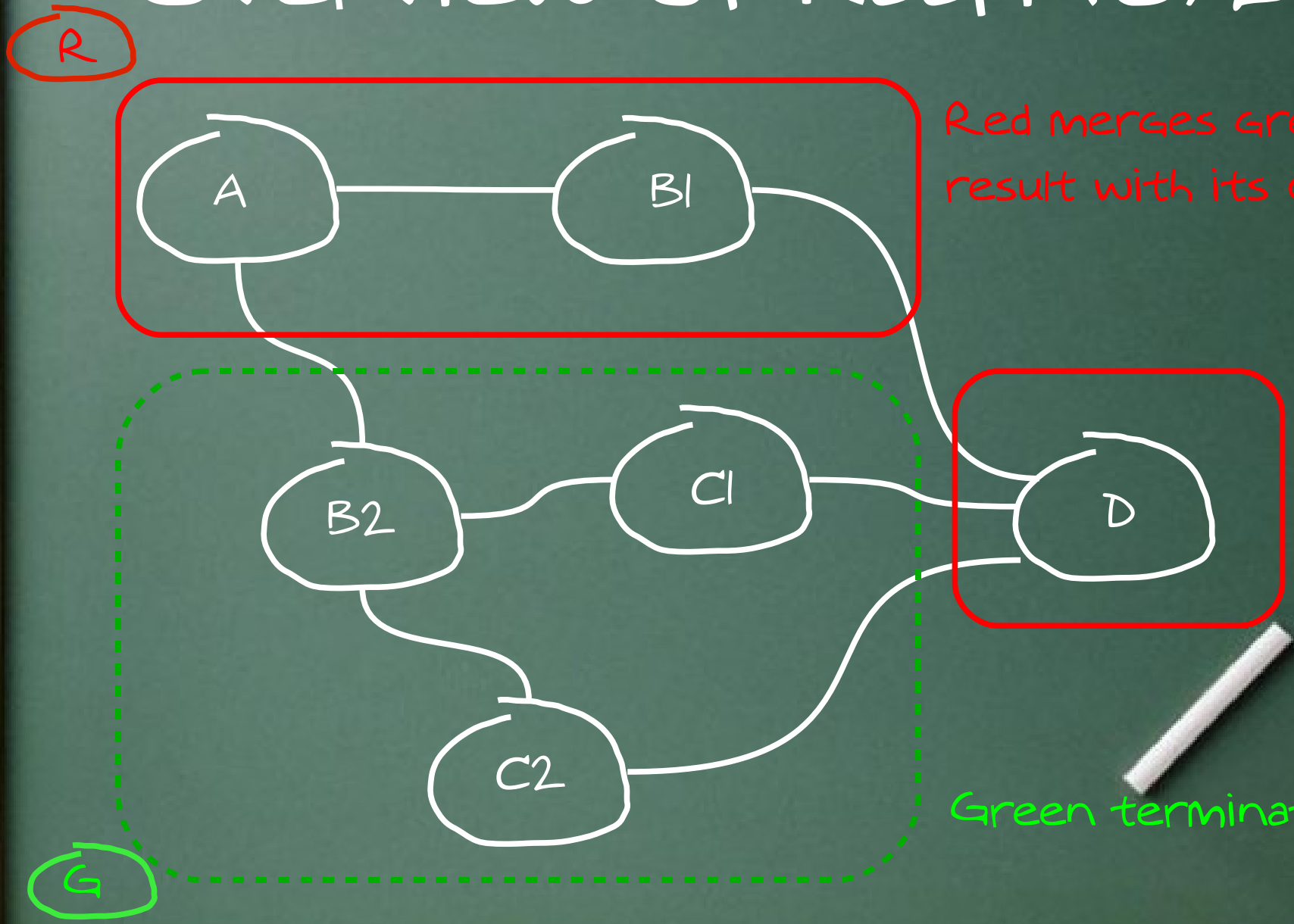


Red follows sequential execution

Which Branch ?

Green follows sequential execution

Overview of Kaapi (5/6)



Overview of Kaapi (6/6)

- Dynamic / static scheduling
- Multiple networks (socknet, taktuk ...)
- Checkpointing (TIC, CCK ...)
- Trace logging
- Remote control
- parallel launcher (taktuk)



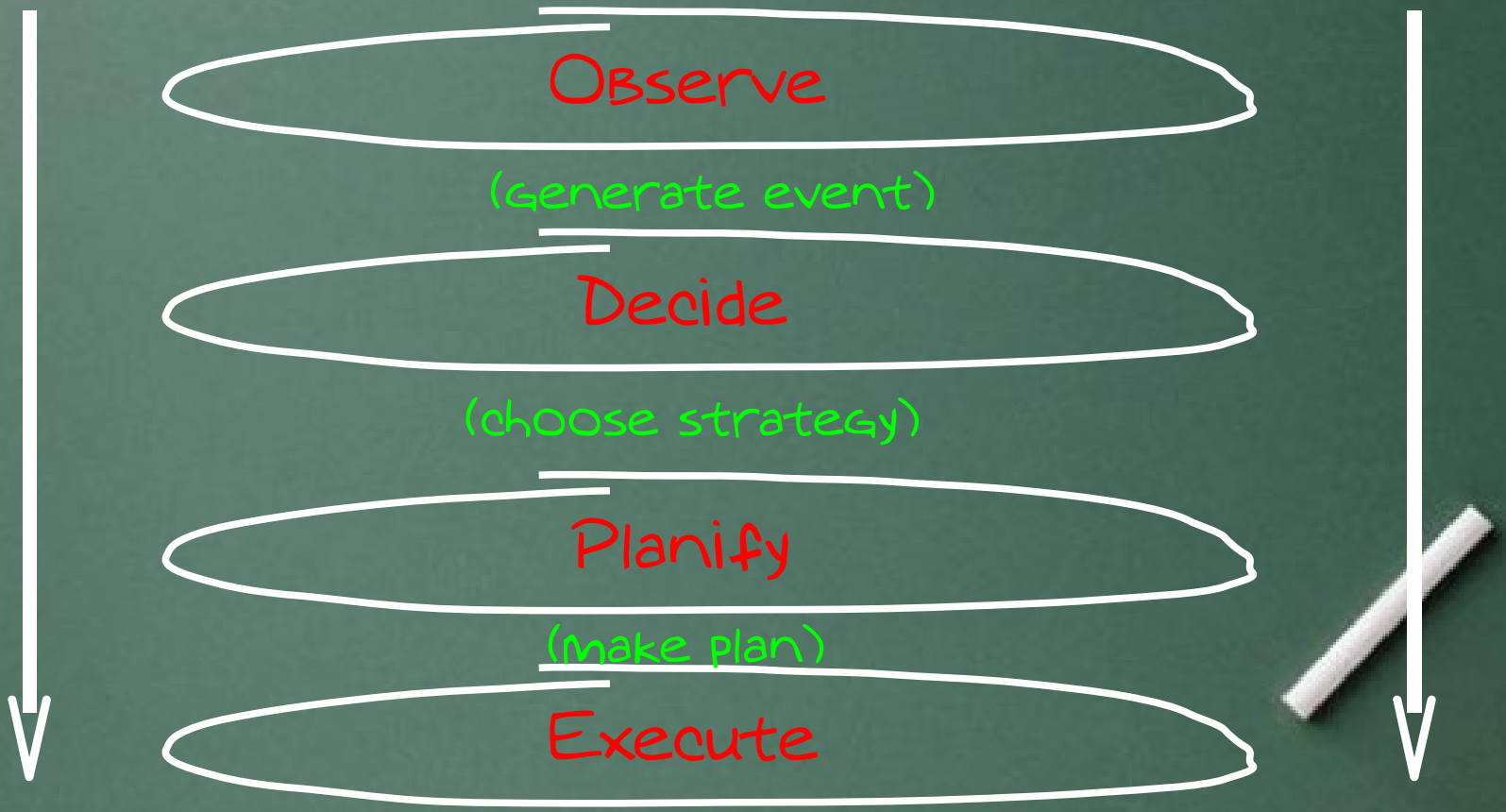
Plan

- ~~Overview of Kaapi~~
- Overview of Dynaco
- Making them work together
- Results
- Conclusion & Future works



Overview of Dynaco (1/5)

- Canvas for the adapted component



Overview of Dynaco (2/5)



un composant (ici commande shell)

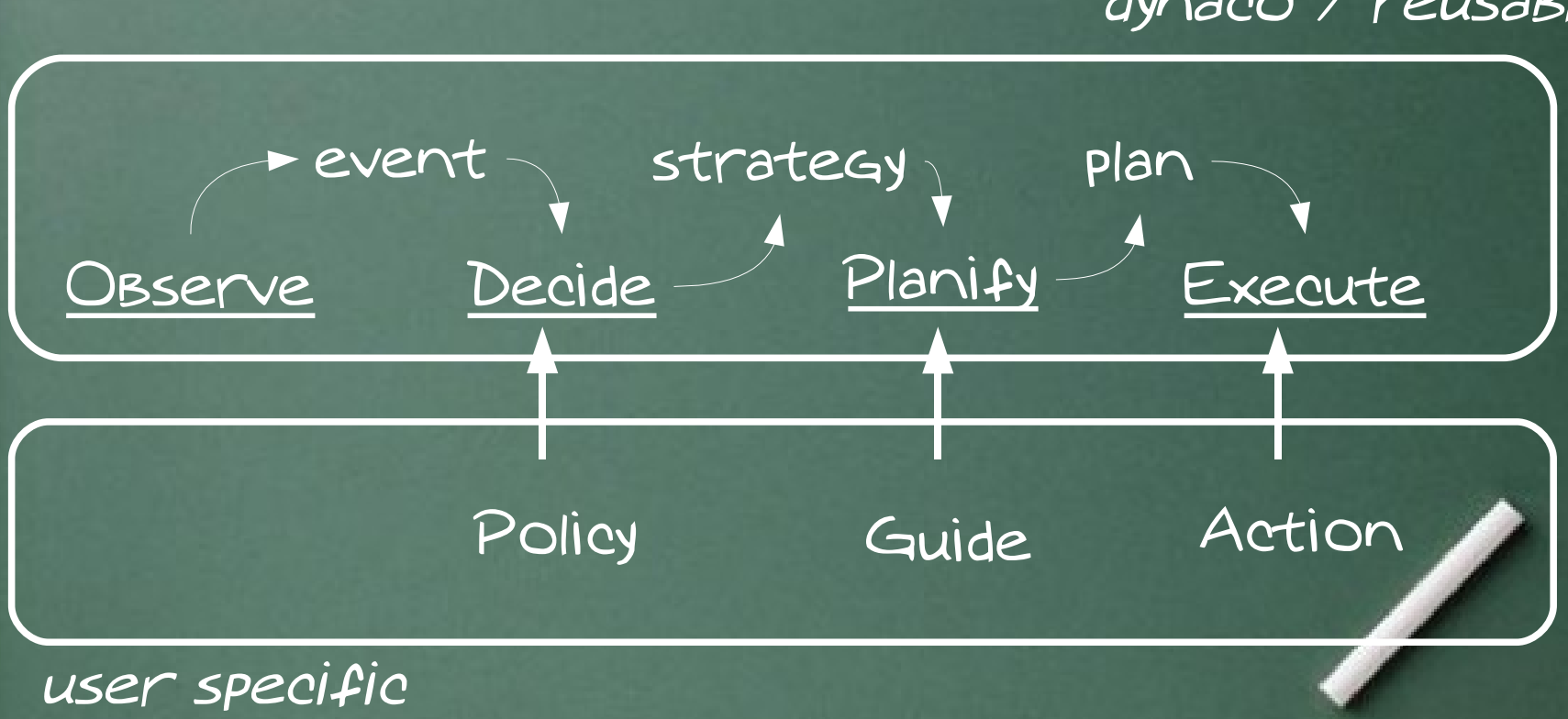
Overview of Dynaco (3/5)

- Model for adaptability
- Propose a canvas to adapt other component
- Based on Fractal (model) & Julia (implem.)
- developed during the thesis of J. Buisson



Overview of Dynaco (4/5)

- Dynaco Architecture



Overview of Dynaco (5/5)

- The adapted component is granted

i. Several probes (get the state)

ii. Several Strategies (how to react)

iii. Several Plans (planify actions)

iv. A controller (apply modif.)

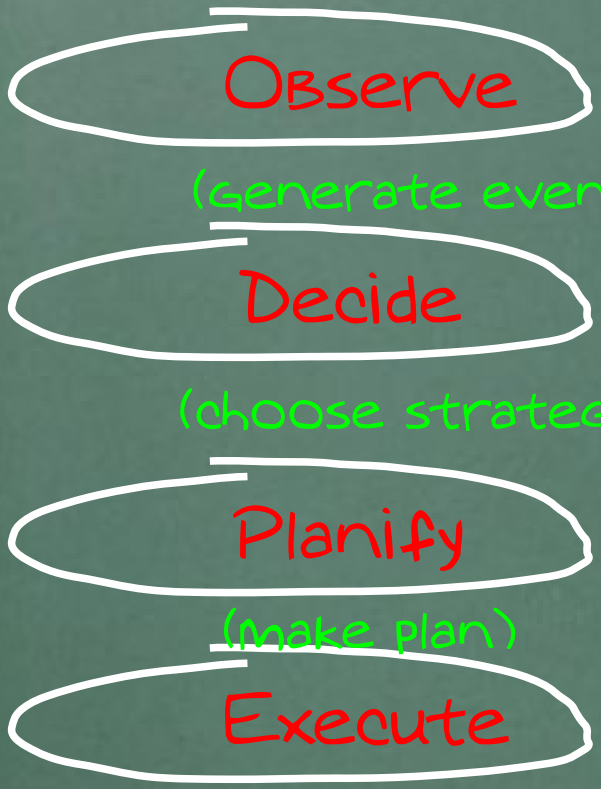


Plan

- ~~Overview of Kaapi~~
- ~~Overview of Dynaco~~
- Making them work together
- Results
- Conclusion & Future works



Making them work together (1/3)



Kaapi Trace Interface

?

Kaapi Control Interface

Making them work together (2/3)

i. Gather information concerning remote steals

ii. If there is a lot of failures, decide to ...

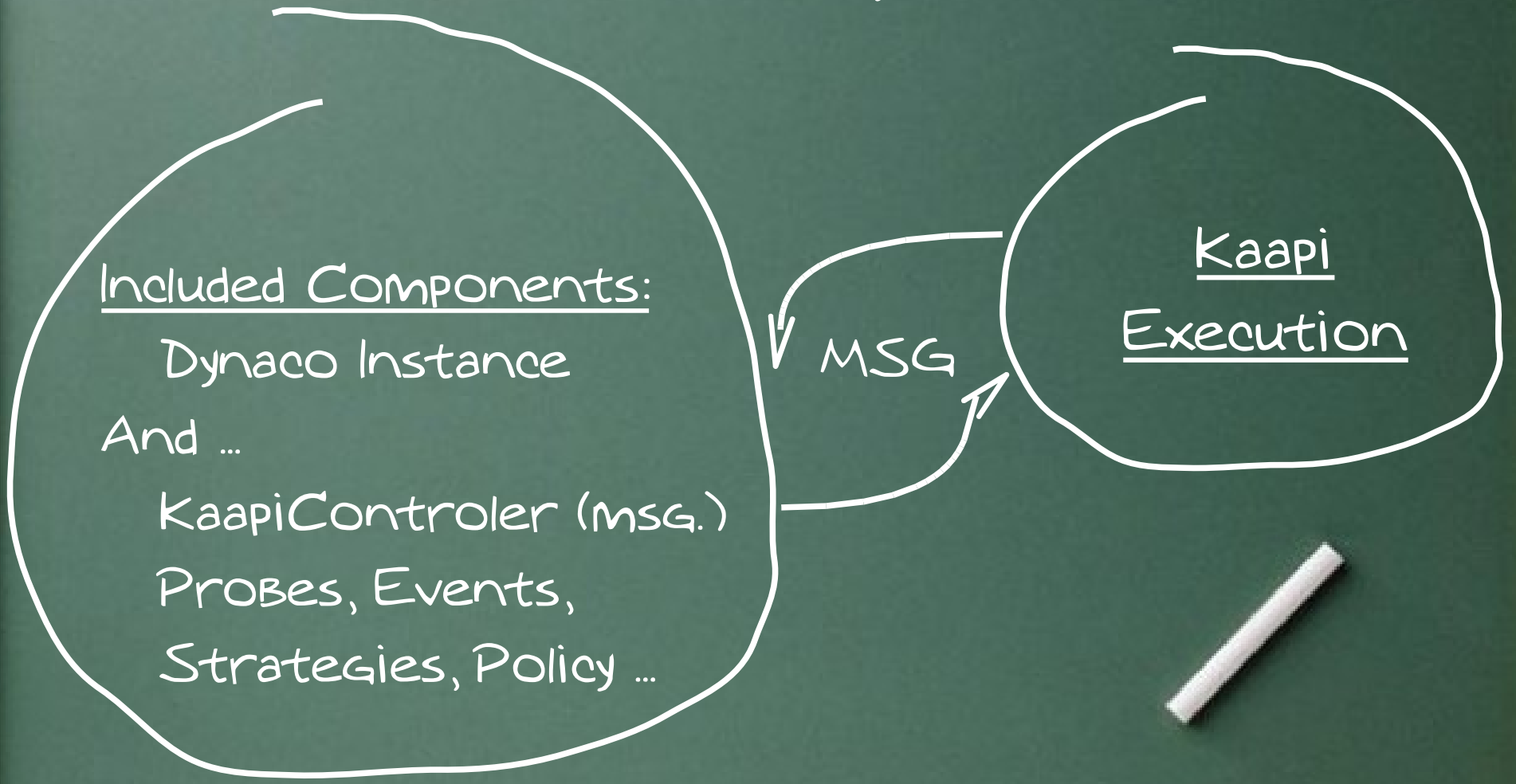
iii. Switch WorkStealing strategy

iv. How ? Broadcast a switch message

v. Do the Broadcast



Making them work together (3/3)



Control Component

Plan

- ~~Overview of Kaapi~~
- ~~Overview of Dynaco~~
- ~~Making them work together~~
- Results
- Conclusion & Future works



Results

- Working Prototype (validates architecture) "comka"
- Wrappers for Kaapi Control in java & perl + ...
- Integration to kaapi's svn
- Still lacks useful strategies



Plan

- ✗ ~~Overview of Kaapi~~
- ✗ ~~Overview of Dynaco~~
- ✗ ~~Making them work together~~
- ✗ ~~Results~~
- ✓ Conclusion & Future works



Conclusion & Future works

- The Cooperation is working well
- Both tools interacts in a fine way !
- Now, please
  - propose strategies for security / safety ?
  - propose probes (internals / externals) ?
  - propose actions ?
- Thank you all !

