L.a.n.c.i.a. RMI

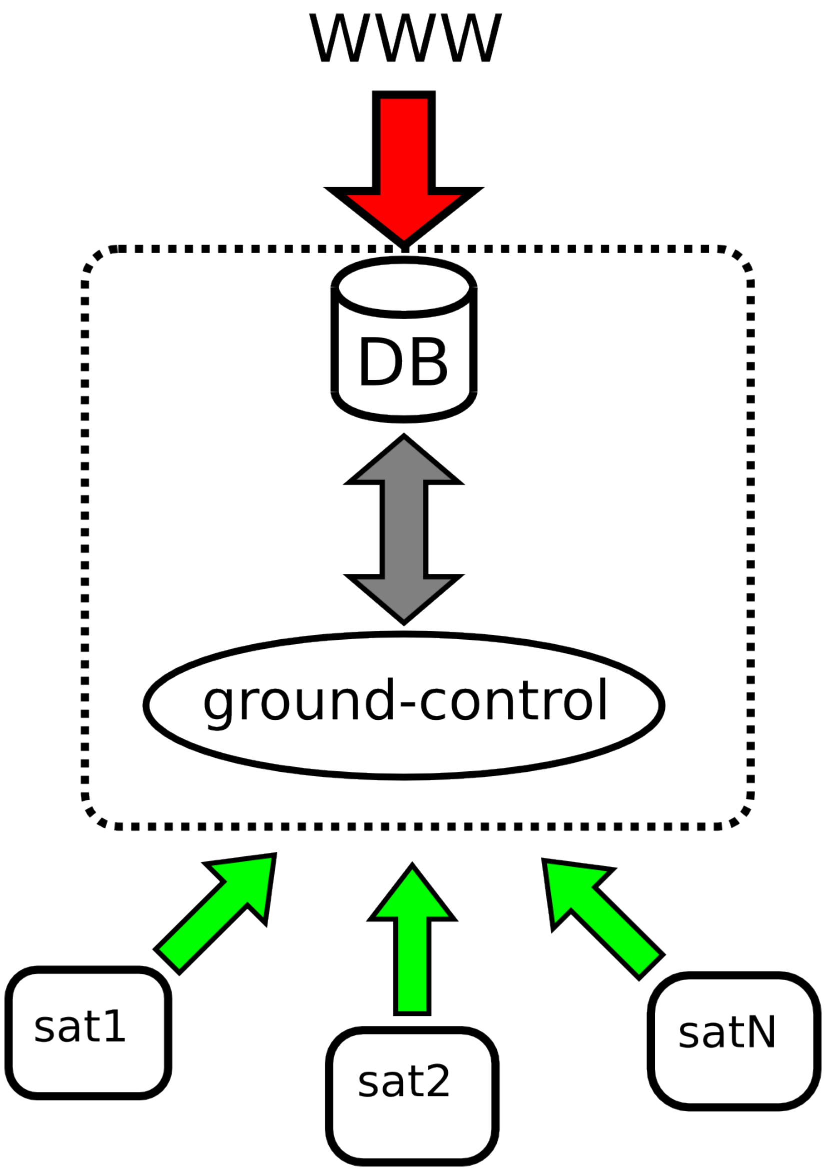
Centralized Advertisement Distribution System built on top of JavaRMI

a.a. 2009/2010

Gaetano Catalli
Matteo Landi
Simone Mainardi

Introduction

- Ground-control
- Satellites

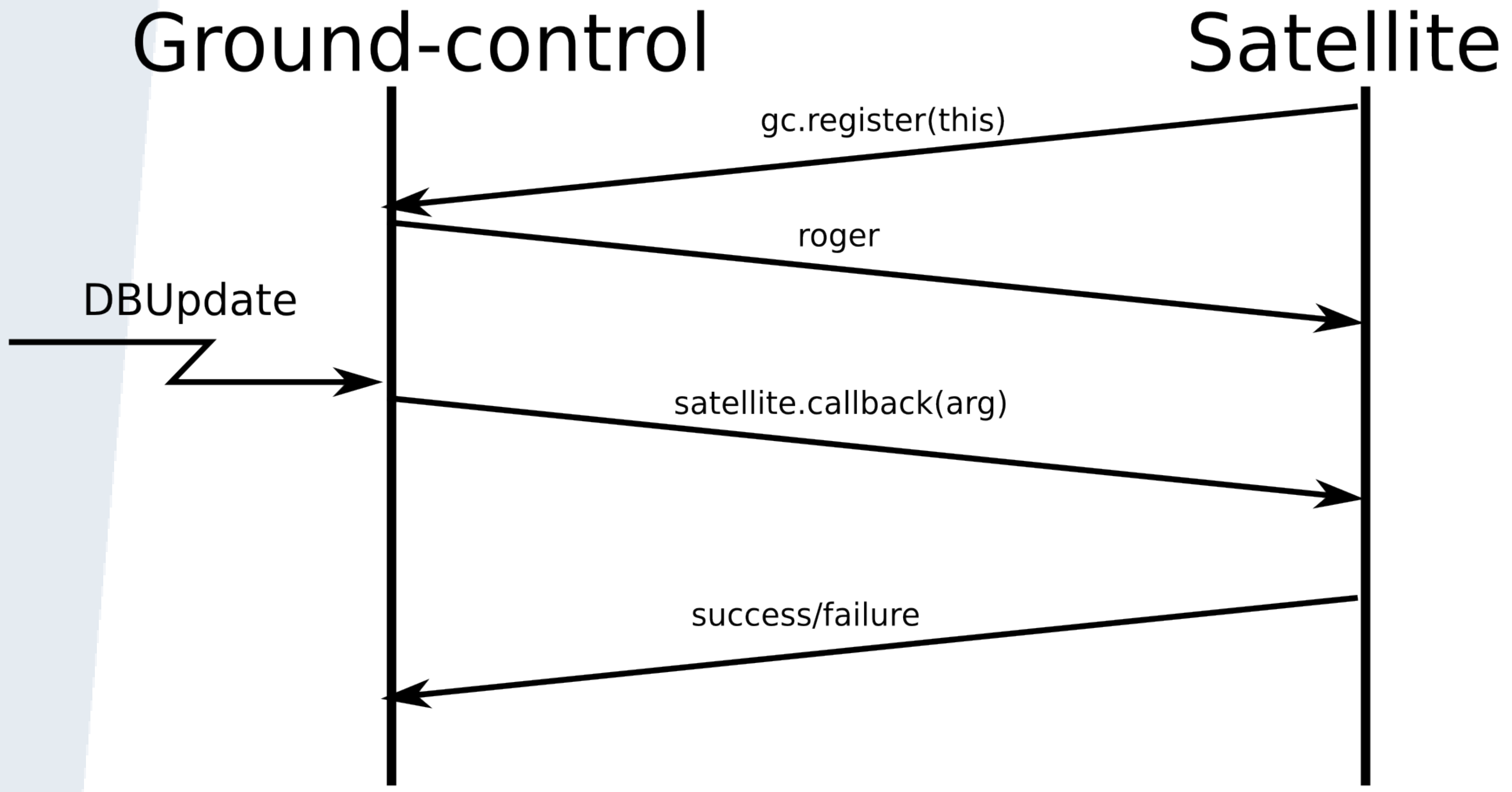


Requirements

- Functional
  - File upload
  - File deletion
  - Open remote secure shell
  - Close remote secure shell
  - Software update
- System
  - Work behind NAT device
  - Security

Implementation

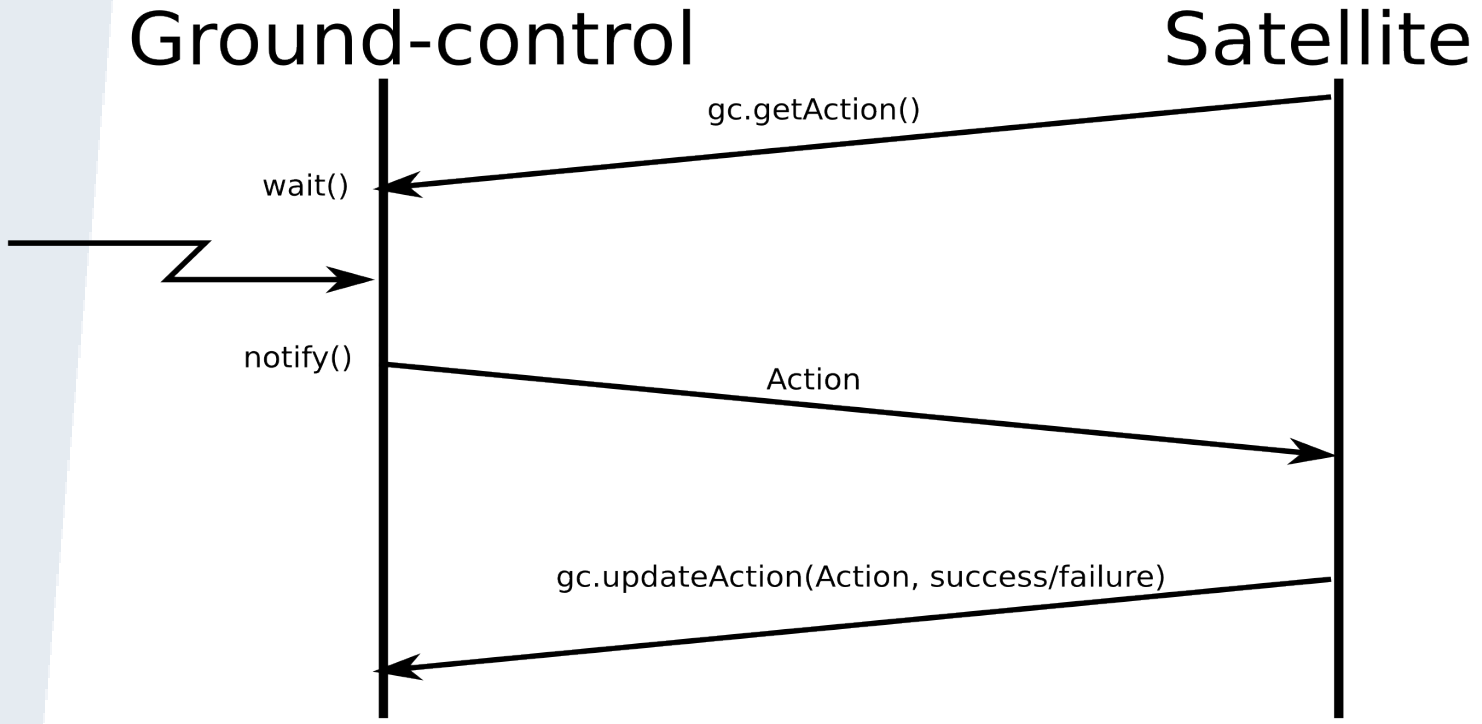
Callbacks ?



Pros: easy to implement, immediate responsivity
Cons: needs of public IP addresses

Implementation (cont...)

Thread pool ?

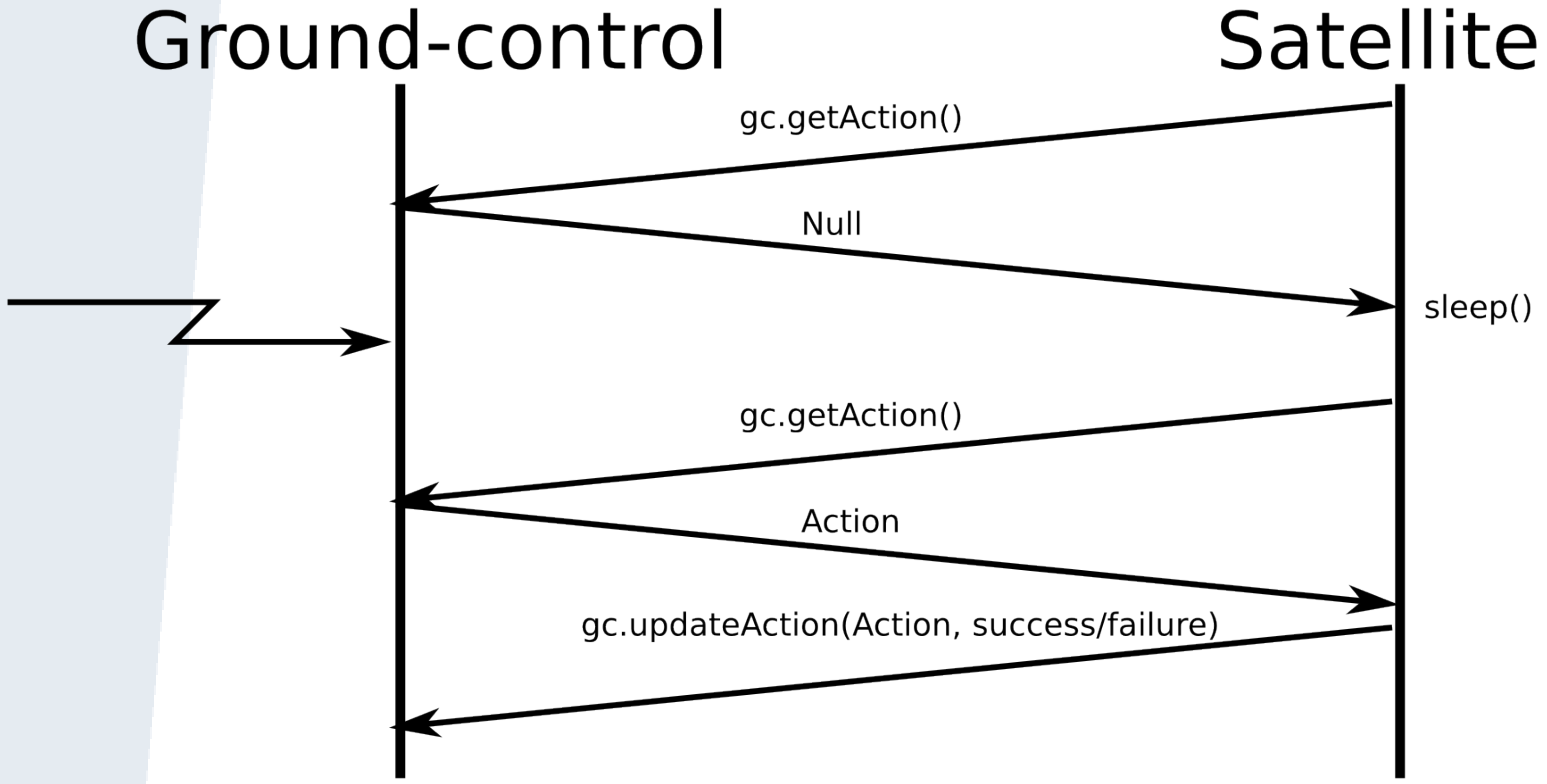


Pros: immediate responsivity, no need of public IP addresses

Cons: thread synchronization, locked threads on satellites unexpected disconnection

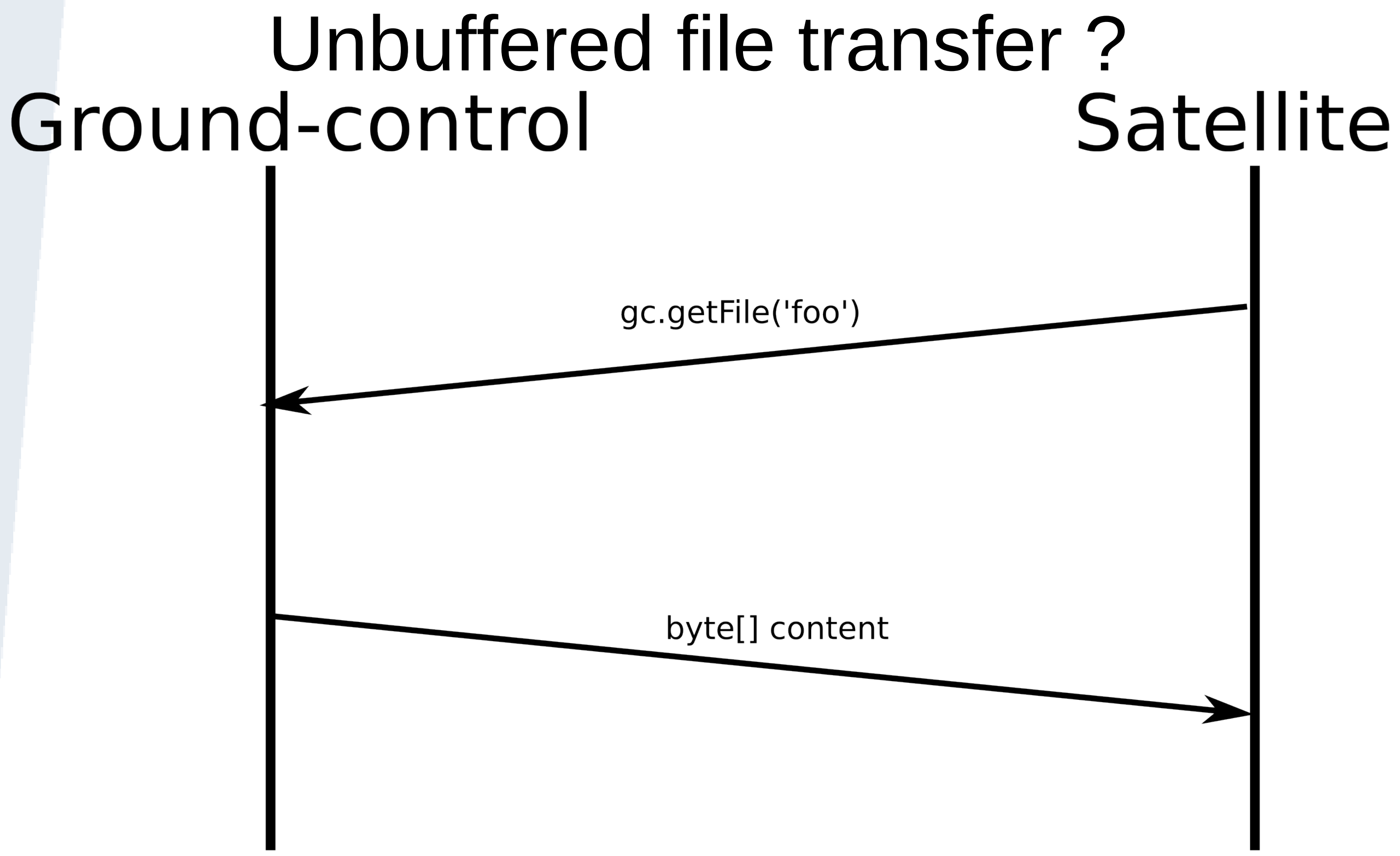
Implementation (cont...)

Polling Satellite !



Pros: easy to implement, no need of public IP addresses
Cons: low but configurable responsivity

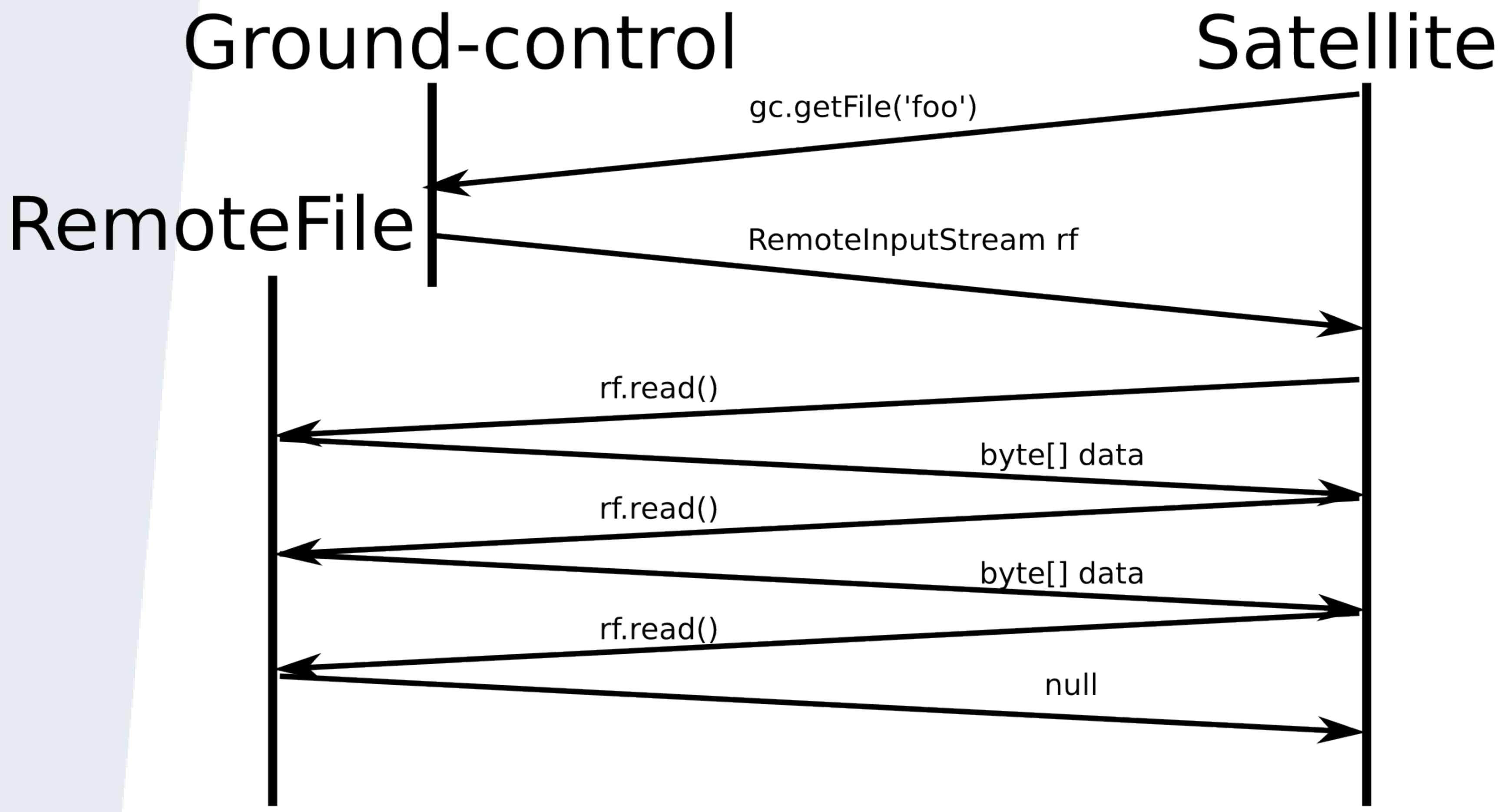
Implementation (cont...)



Pros: easy implementation, minimum overhead
Conts: high memory requirements

Implementation (cont...)

Buffered file transfer !



Pros: optimized memory utilization
Conts: time overhead due to communication RTT

Implementation

Security

- Secure Socket Layer (*javax.rmi.ssl.**)
    - SslRMIClientSocketFactory
    - SslRMIServerSocketFactory

Ground-control trusts each satellite.
Each Satellite trust ground-control.

Distribution

```
bin/
|-- run-groundcontrol.sh
`-- run-satellite.sh
keystores/
Makefile
policy/
src/
|-- GroundControlApp.java
|-- lancia
|   |-- groundcontrol
|   |   |-- database
|   |   |   |-- Action.java
|   |   |   `-- DatabaseHandler.java
|   |   |-- GroundControlImpl.java
|   |   `-- GroundControl.java
|   `-- rio
|       |-- RemoteInputStreamImpl.java
|       `-- RemoteInputStream.java
`-- SatelliteApp.java
```