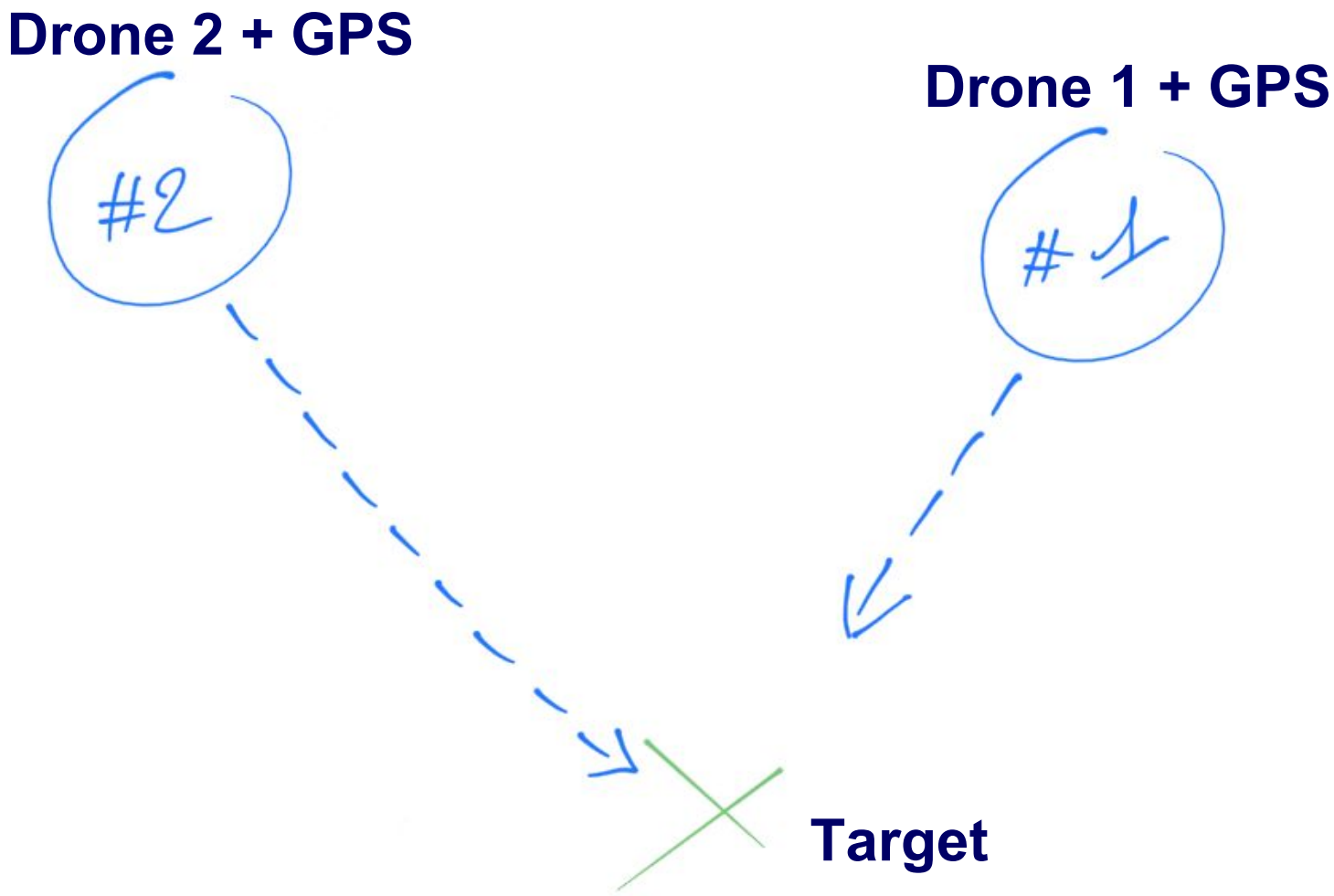
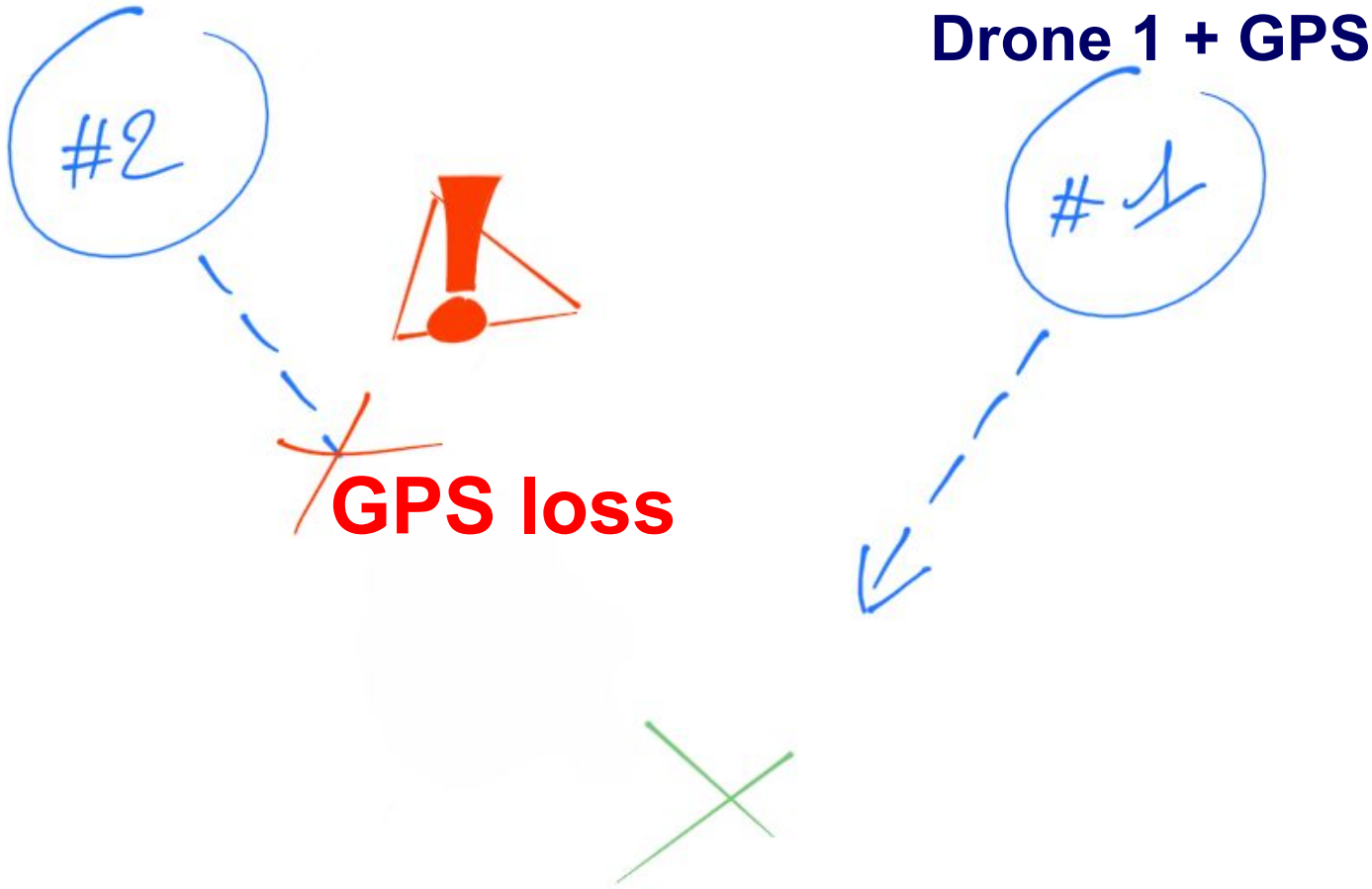
Lub
A language for Dynamic Context Oriented Programming

Steven Costiou
Mickaël Kerboeuf, Glenn Cavarlé, Alain Plantec
UMR CNRS 6285, Lab-STICC/MOCS
Université de Bretagne Occidentale


Drone 2 + GPS
#2
Drone 1 + GPS
#1
Target

Drone 2 **NO GPS**



Unanticipated

Unanticipated

=

Not foreseen at design time

Unanticipated

=

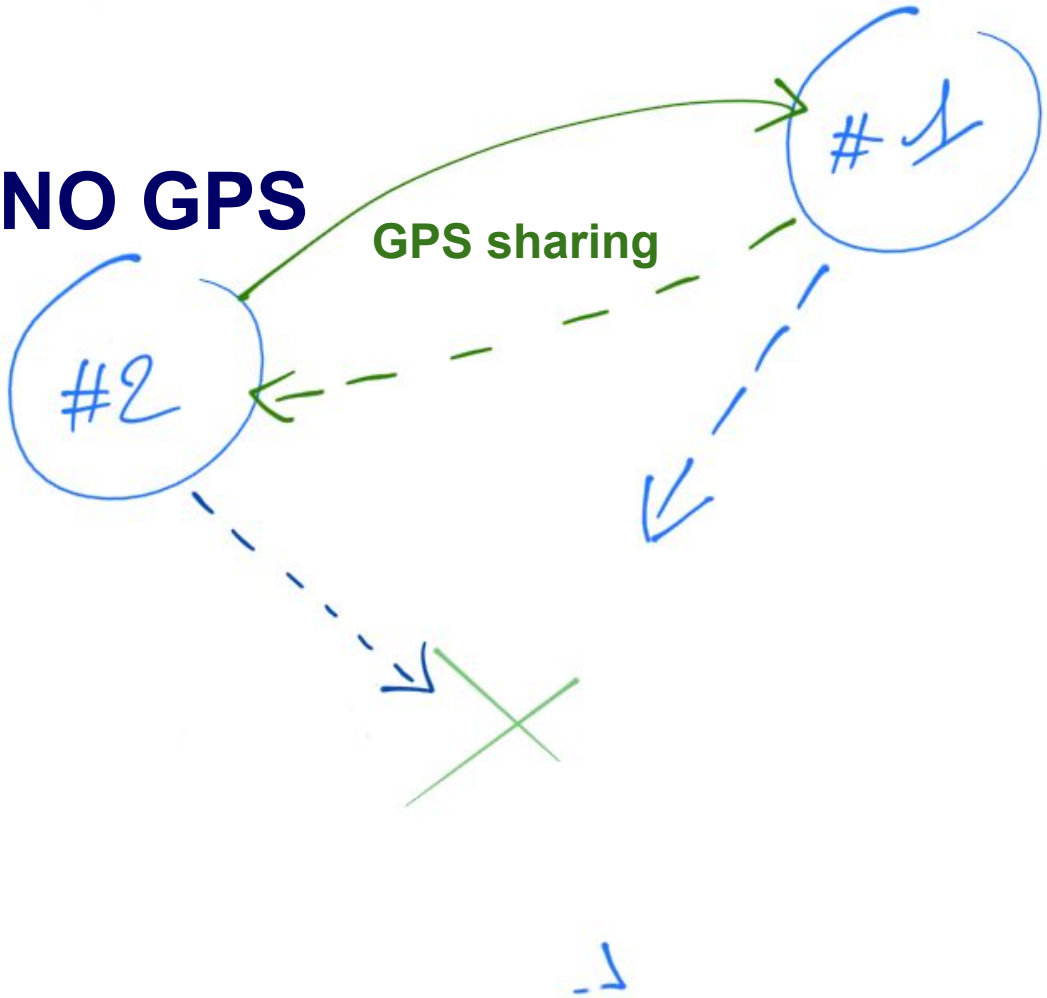
Not foreseen at design time

=

The behavior we need was not expected

Drone 1 + GPS

Drone 2 **NO GPS**



Unanticipated behavior adaptation at runtime

Minimal anticipation

Minimal anticipation

=

Anticipate the need to face unanticipated adaptation

Minimal anticipation

=

Anticipate the need to face unanticipated adaptation

=

Build Support (frameworks)

Dynamic context oriented programming (COP)

What we adapt: Objects

- Instance based adaptation
- Adapted object preserve their identity

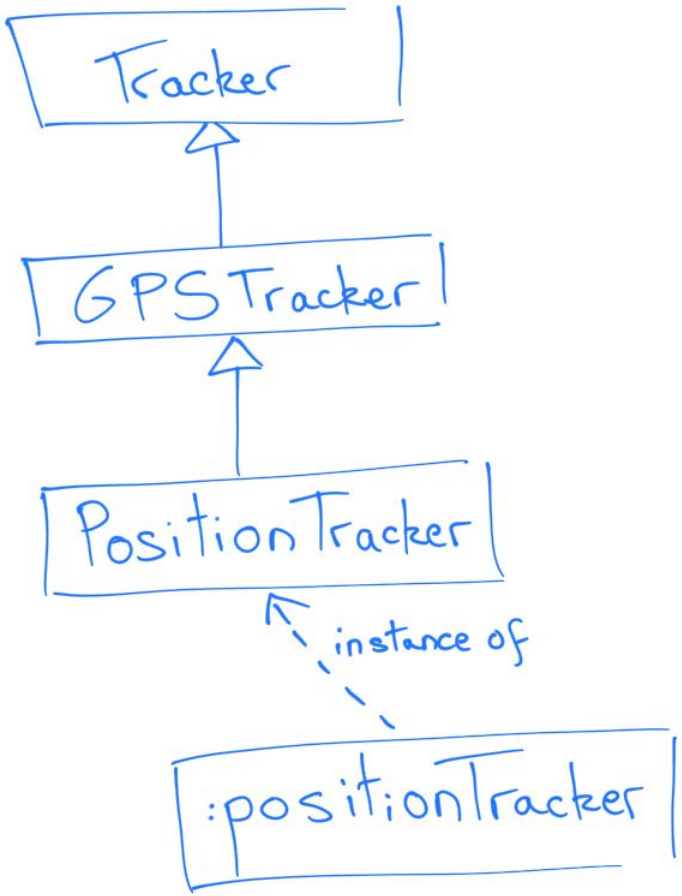
How do we adapt ?

- Free adaptation strategy
- Easy and controlled behavior selection

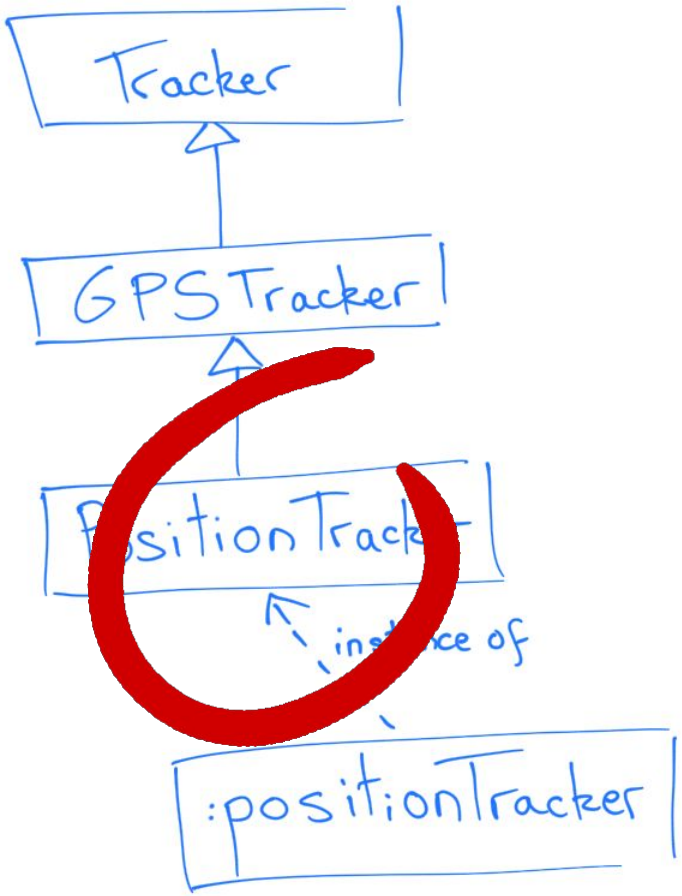
LUB

An extension of Pharo

Behavior adaptation through dynamic lookup control


Tracker
GPSTracker
PositionTracker
instance of
:positionTracker

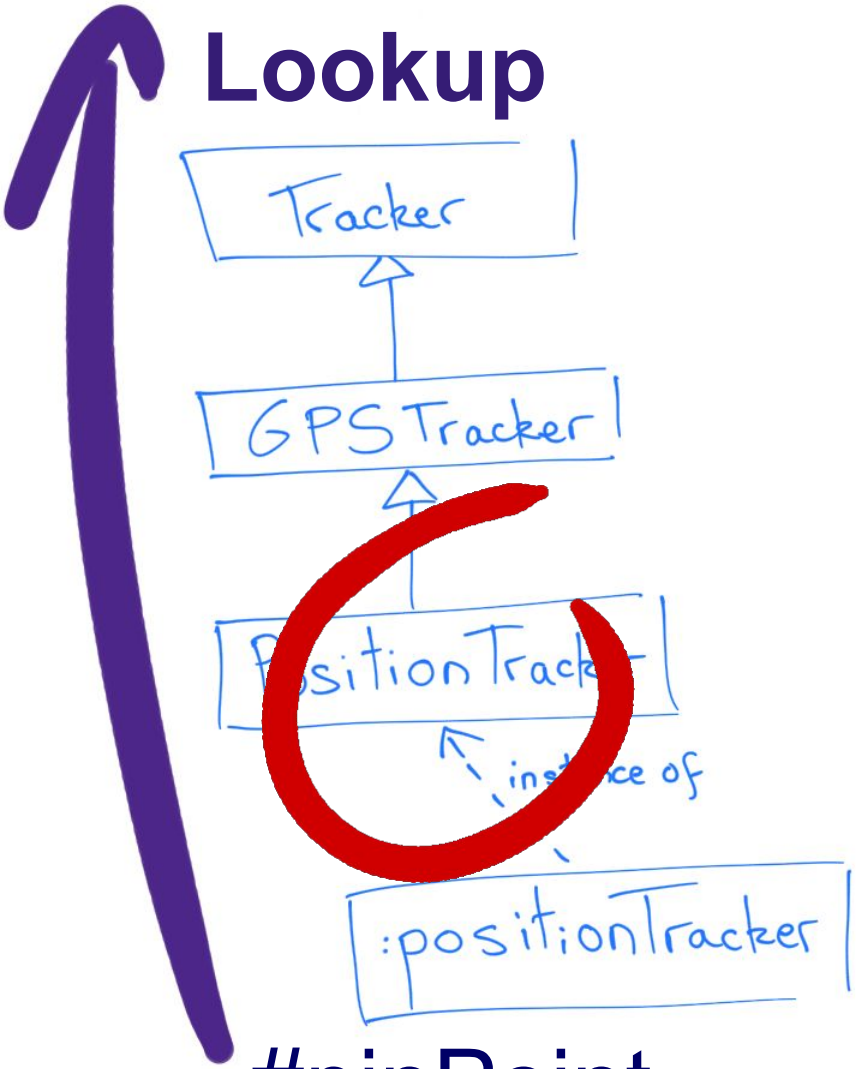
#pinPoint



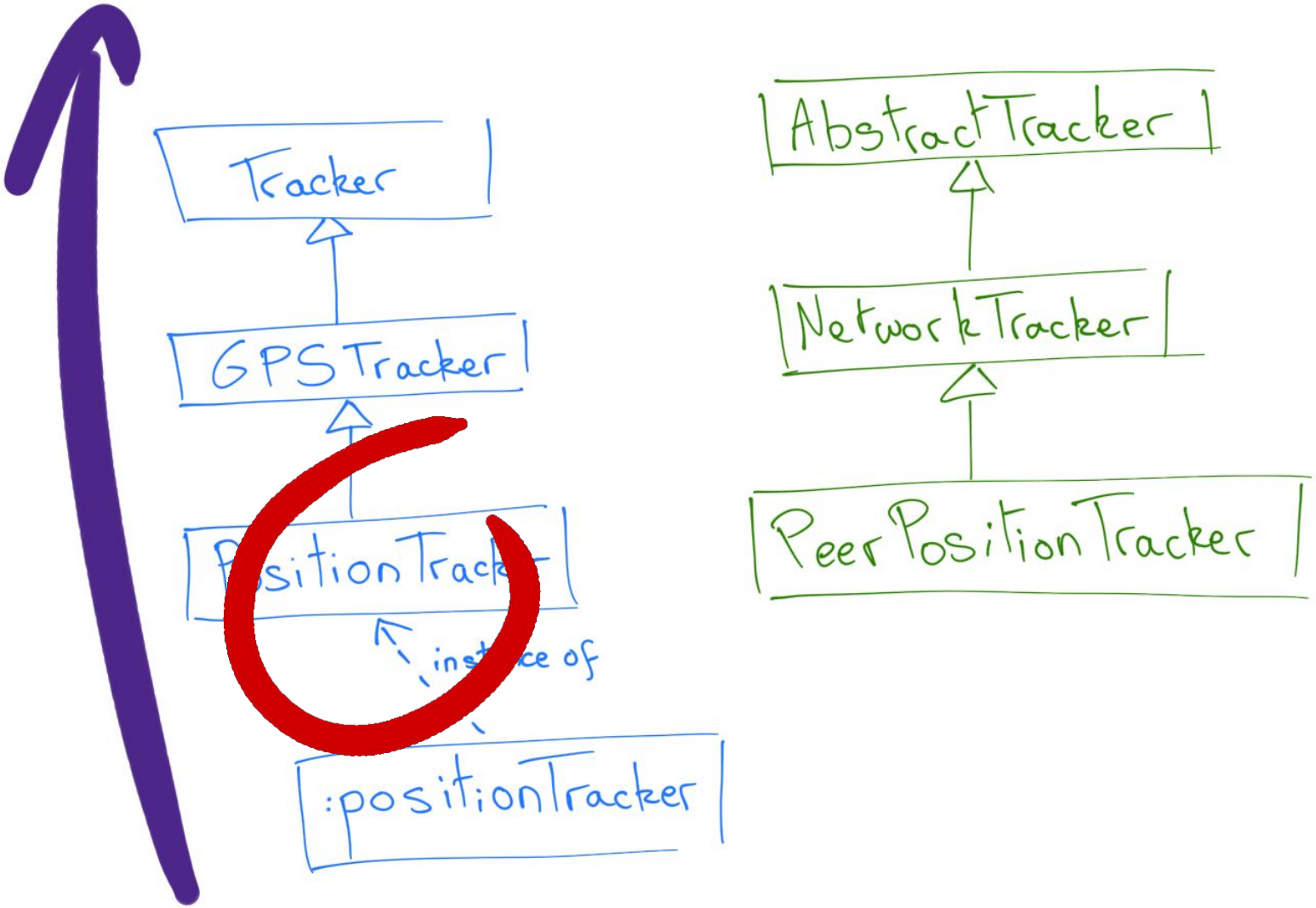
Lookup base

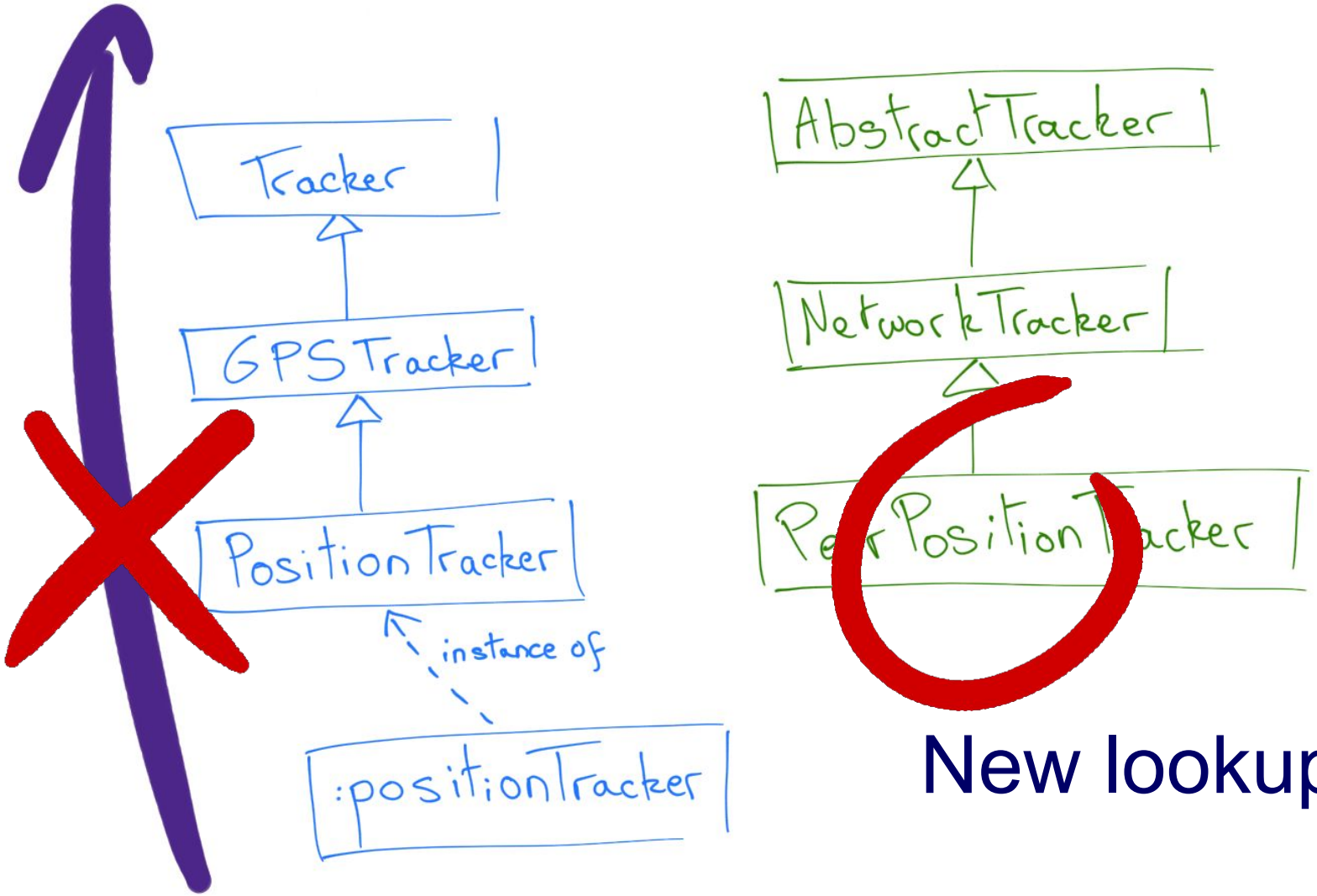
#pinPoint

Lookup



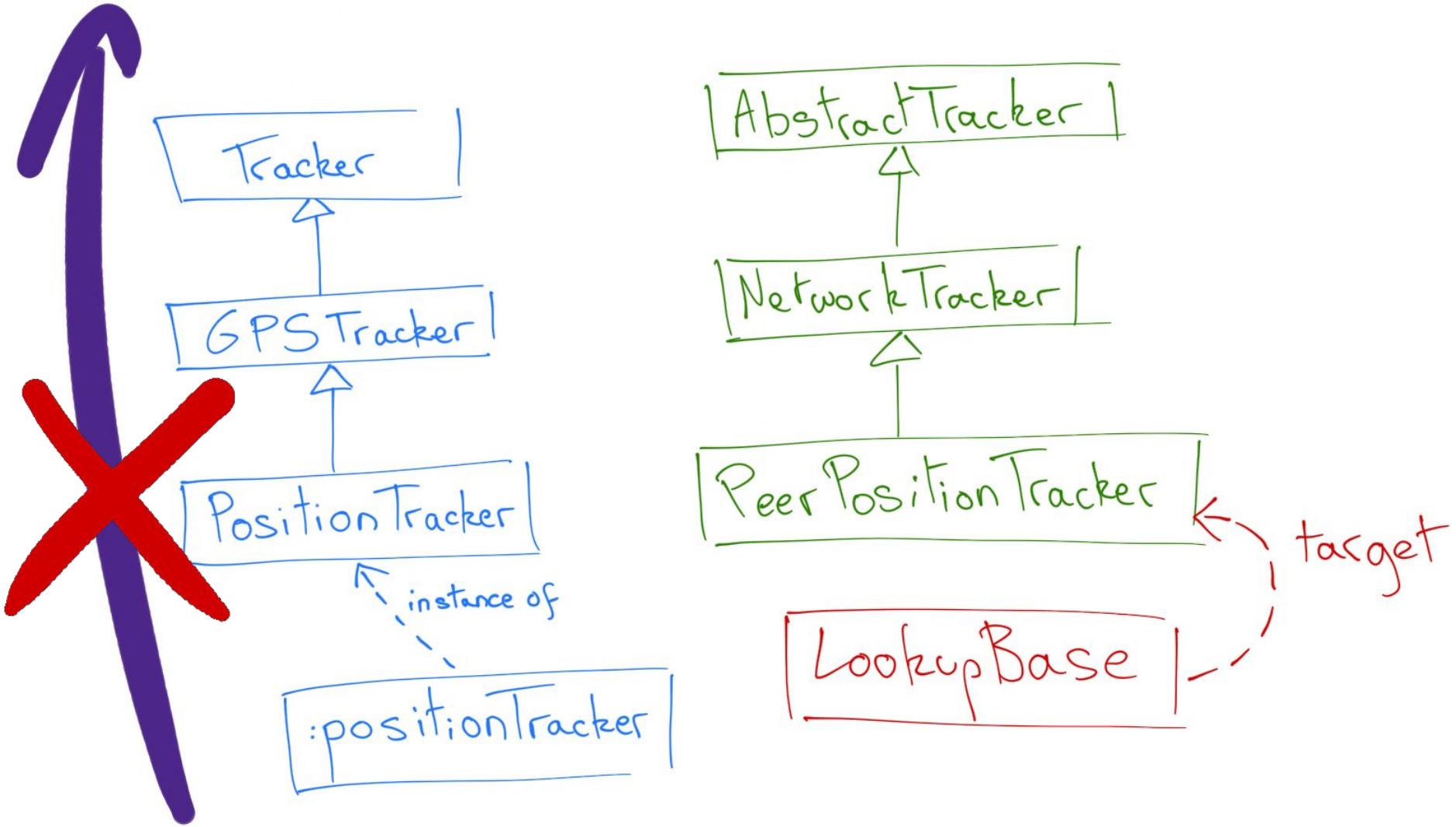
Lookup starts in the object's lookup base

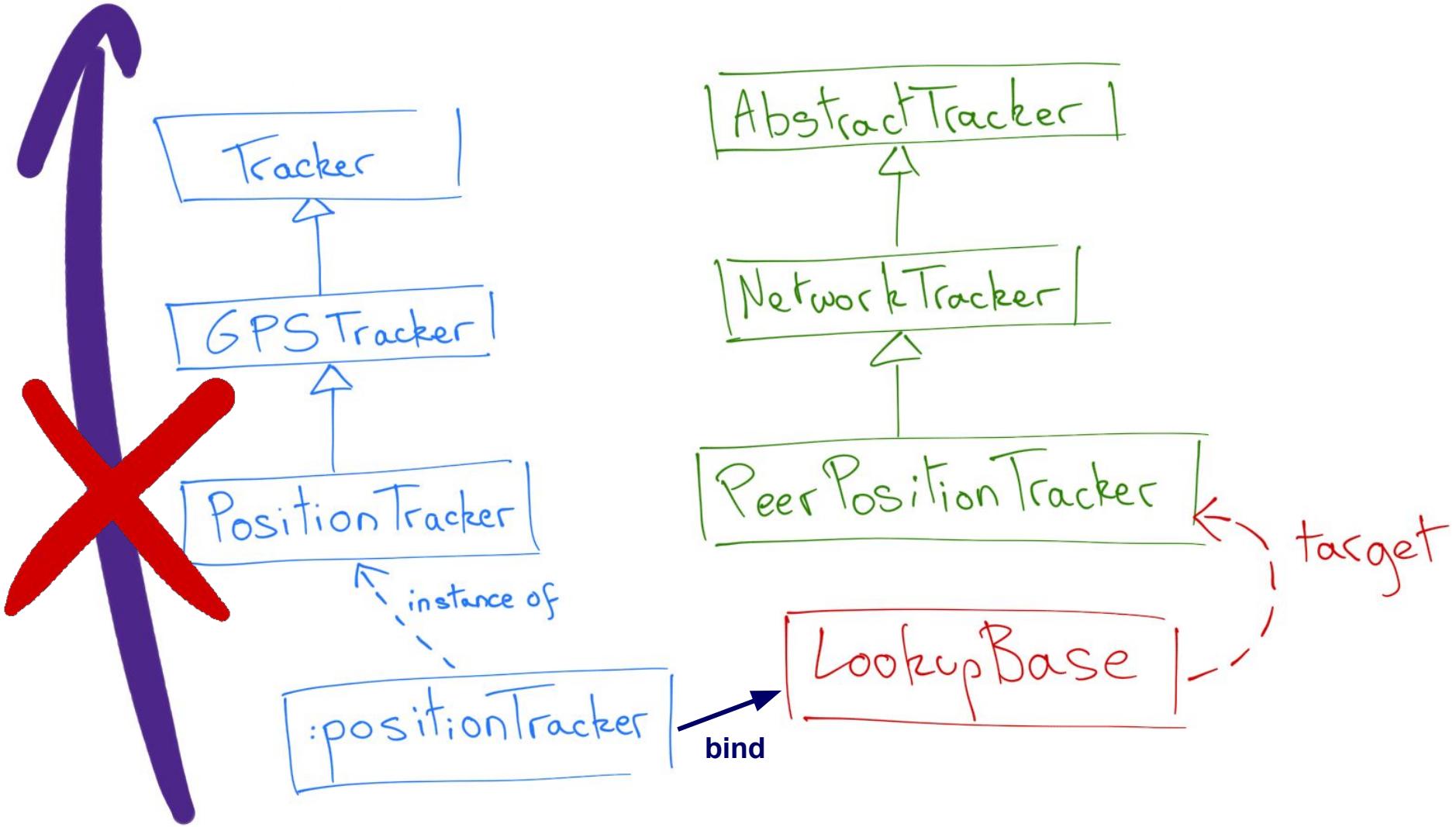

Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
AbstractTracker
NetworkTracker
PeerPositionTracker



New lookup base

```
lub := LookupBase
        named: 'PeerTrackerLookupBase'
        targetClass: PeerPositionTracker
```

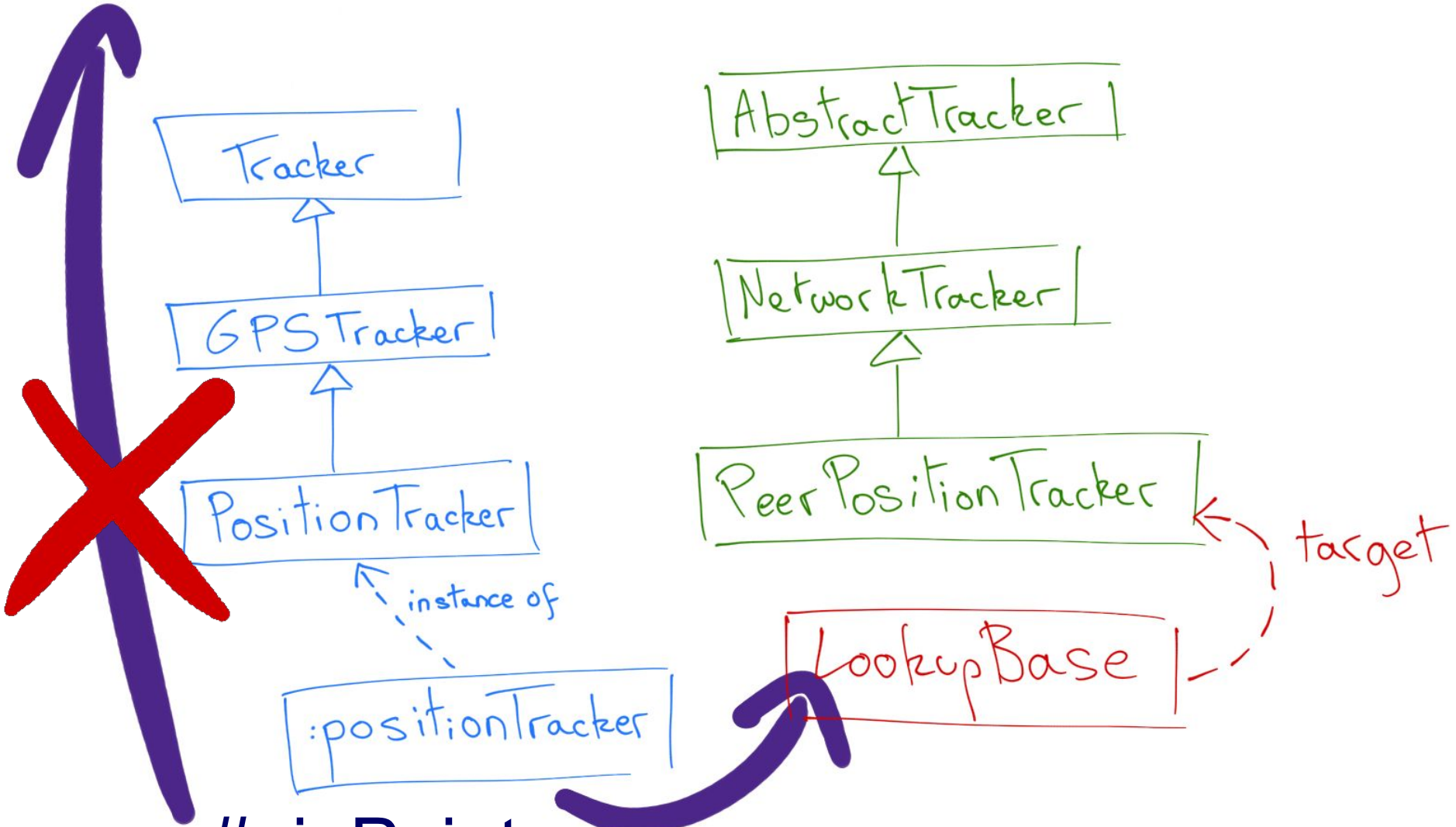

Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
AbstractTracker
NetworkTracker
PeerPositionTracker
target
LookupBase


Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
bind
AbstractTracker
NetworkTracker
PeerPositionTracker
target
LookupBase

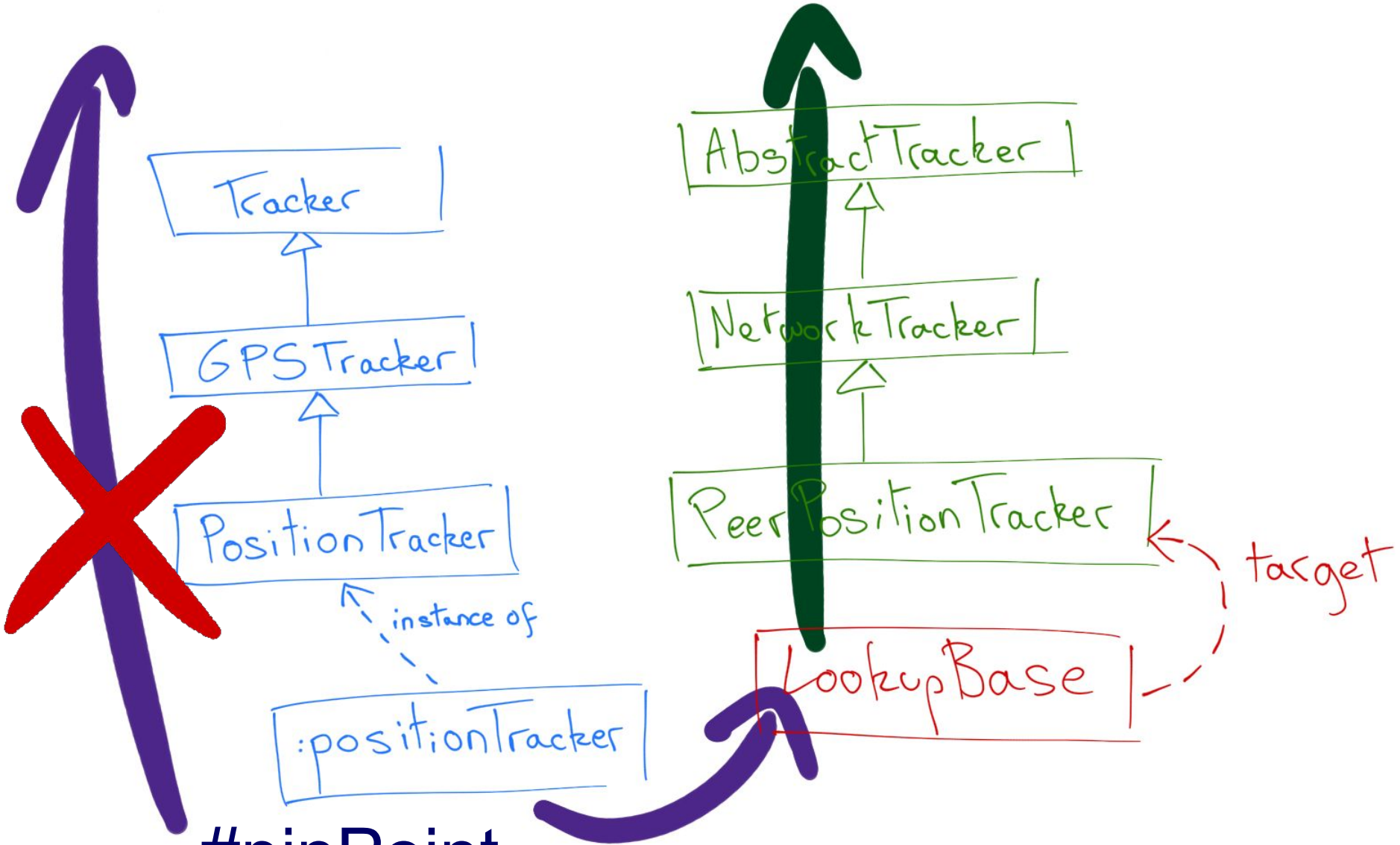
```
lub := LookupBase
        named: 'PeerTrackerLookupBase'
        targetClass: PeerPositionTracker
```

positionTracker **lookupBase:** lub.

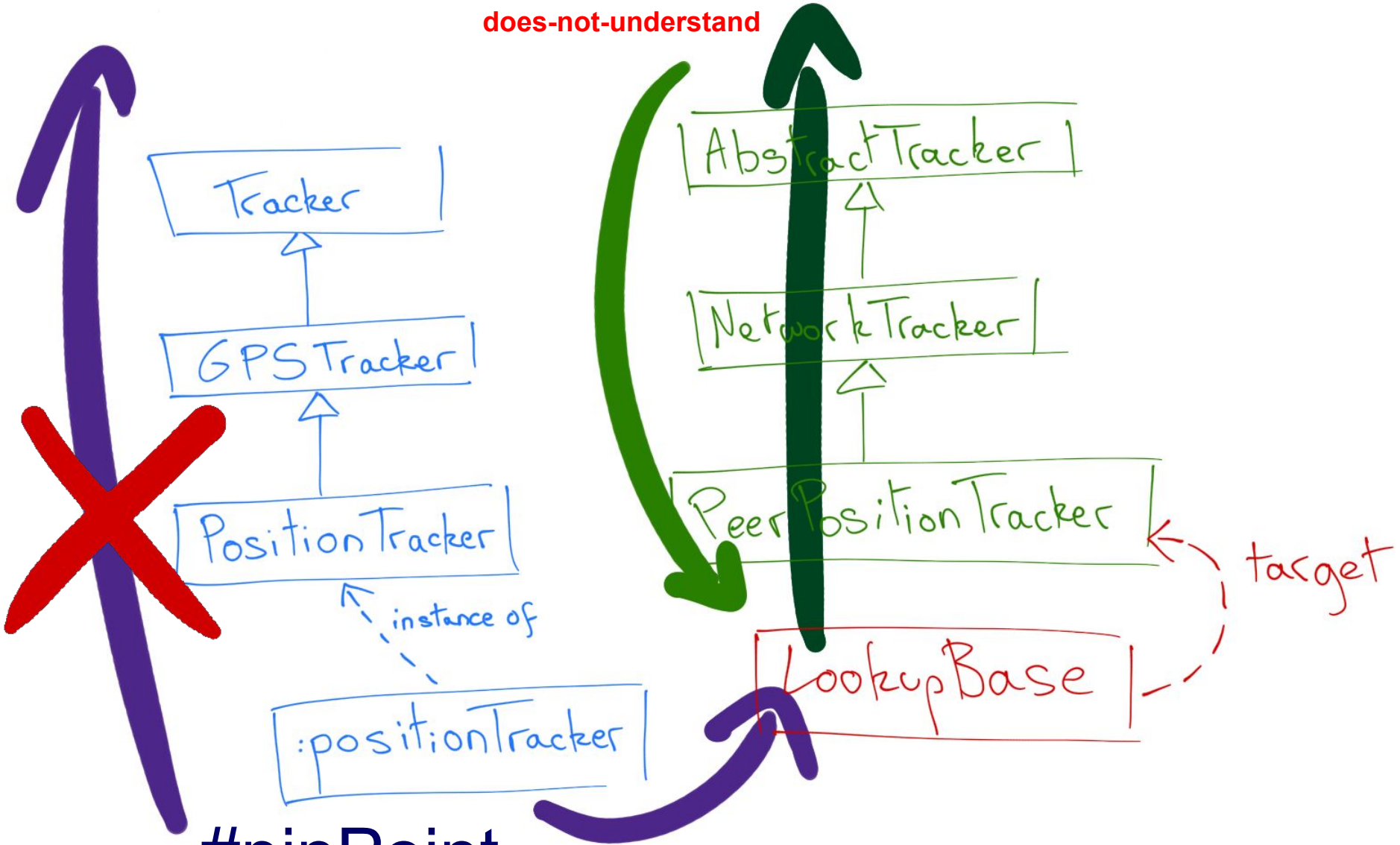
positionTracker pinPoint


Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
AbstractTracker
NetworkTracker
PeerPositionTracker
target
LookupBase

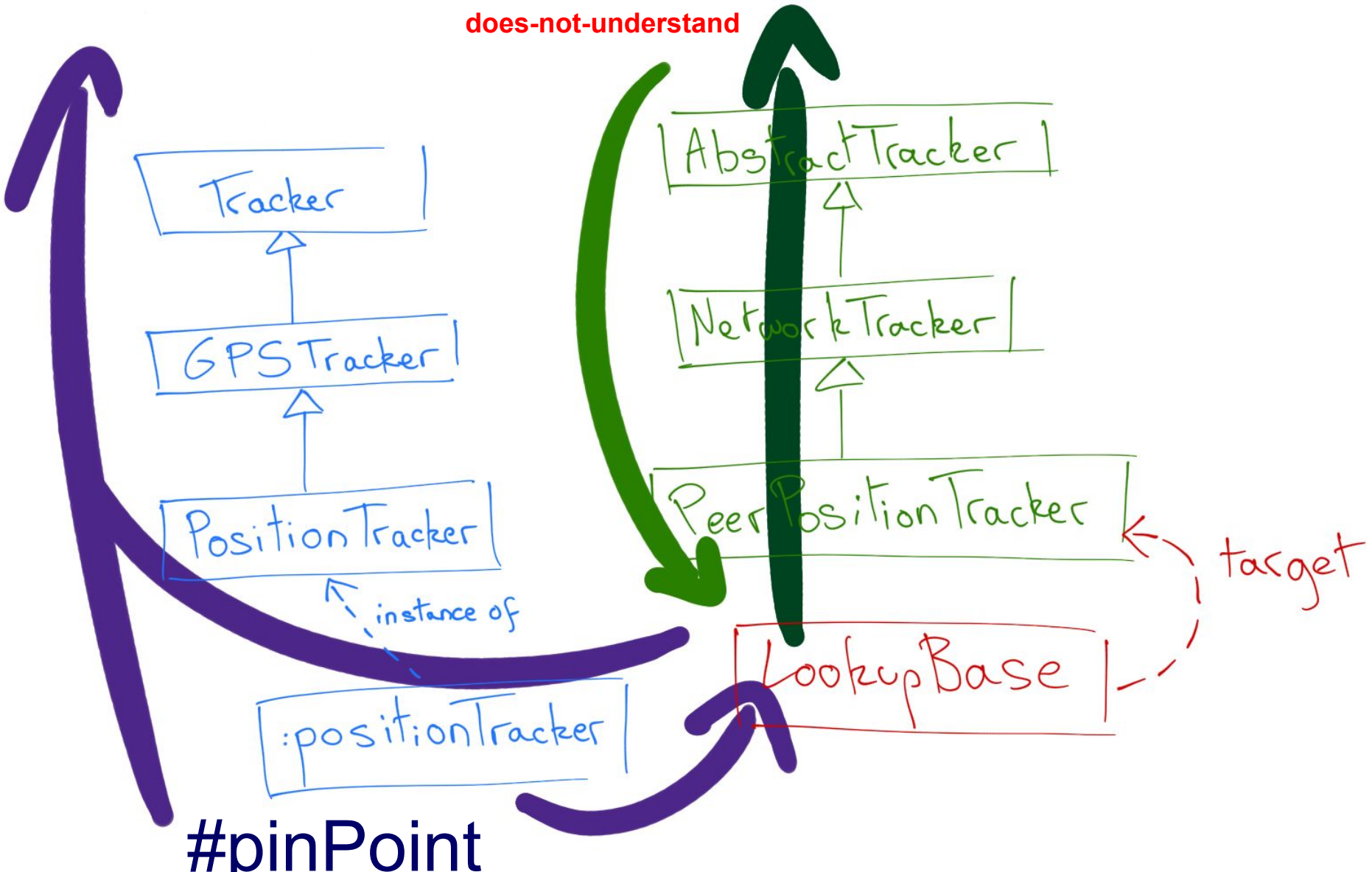
#pinPoint


Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
AbstractTracker
NetworkTracker
PeerPositionTracker
LookupBase
target

#pinPoint


does-not-understand
Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
AbstractTracker
NetworkTracker
PeerPositionTracker
LookupBase
target

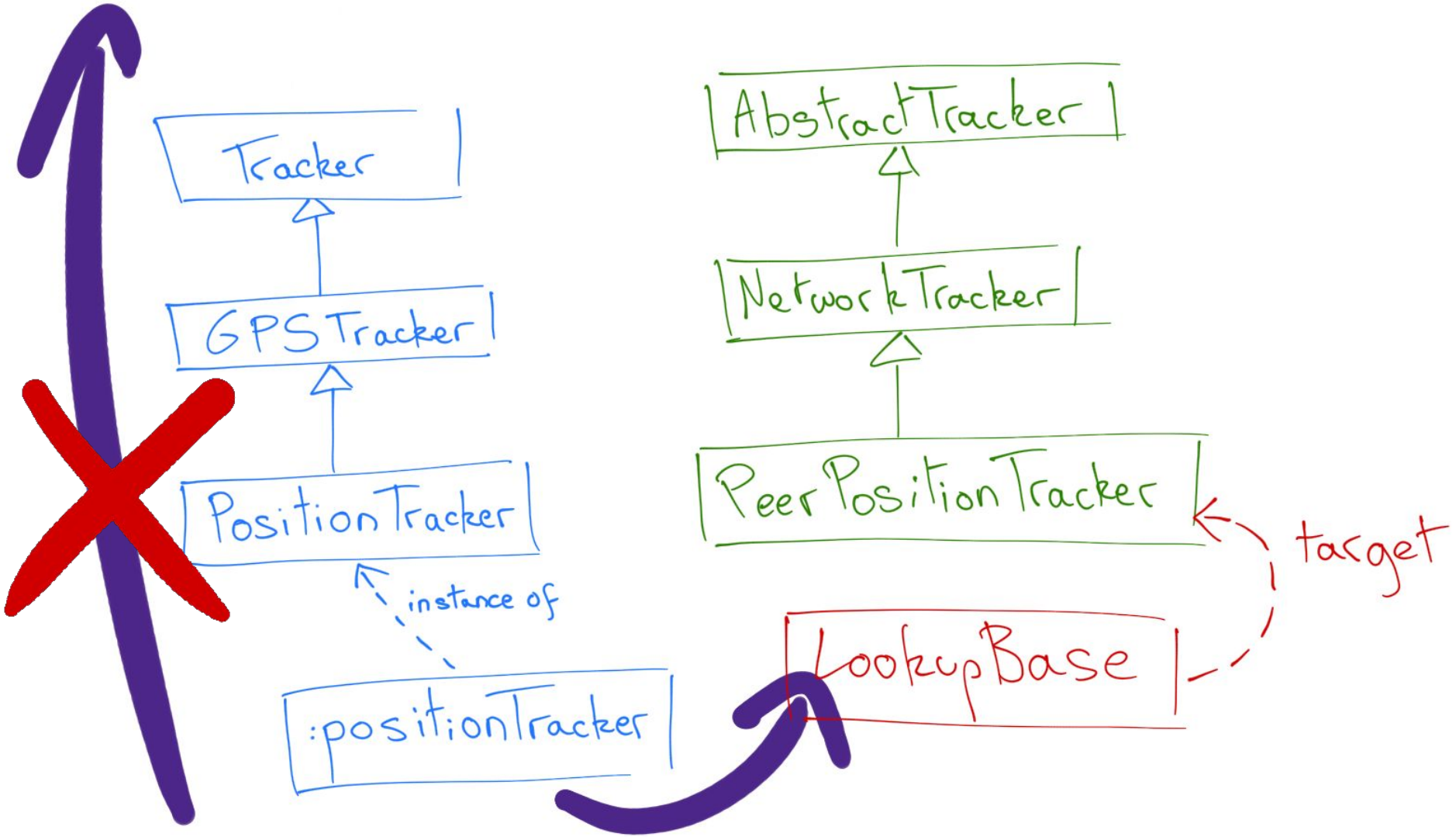
#pinPoint

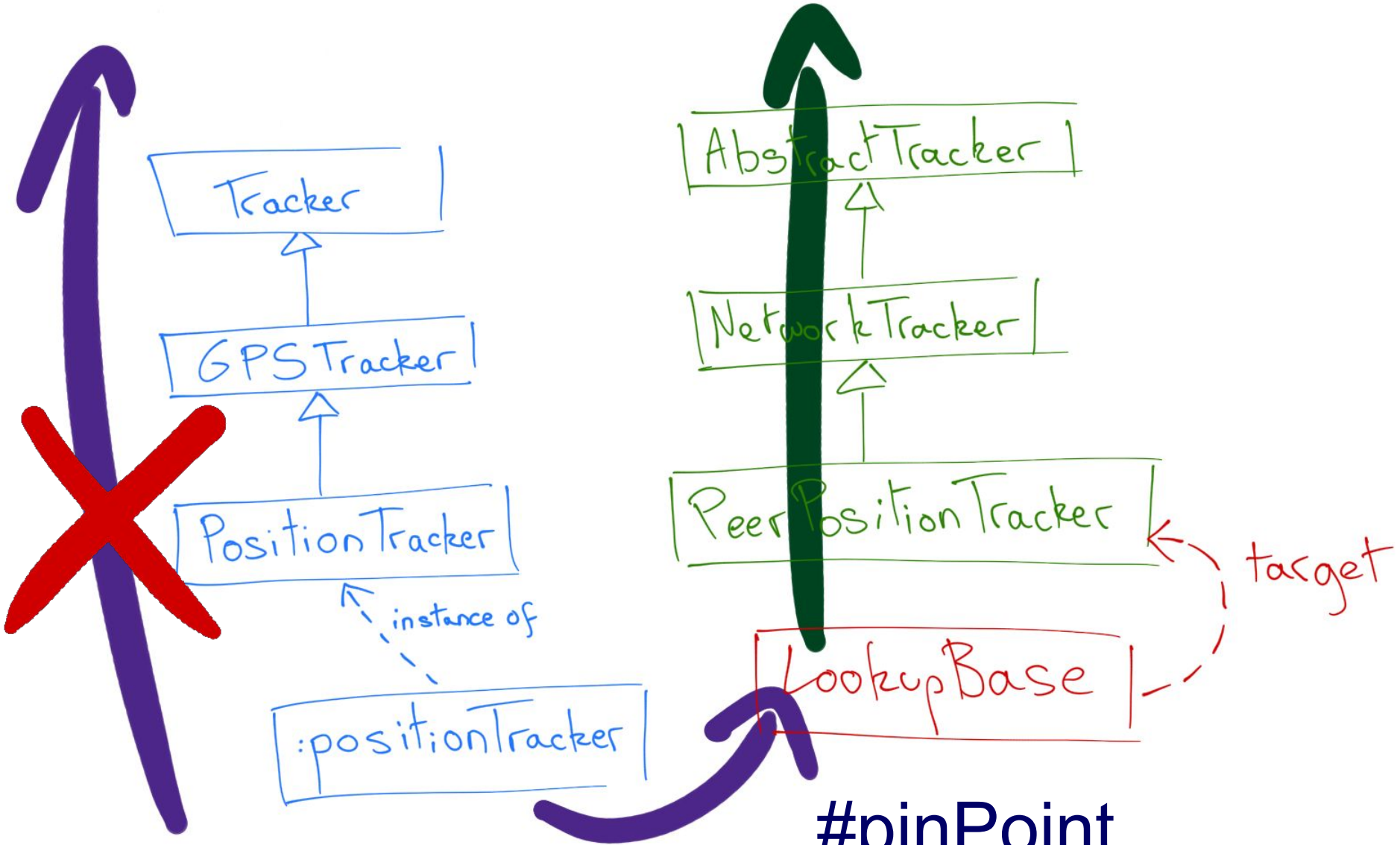

does-not-understand
Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
AbstractTracker
NetworkTracker
PeerPositionTracker
LookupBase
target

#pinPoint

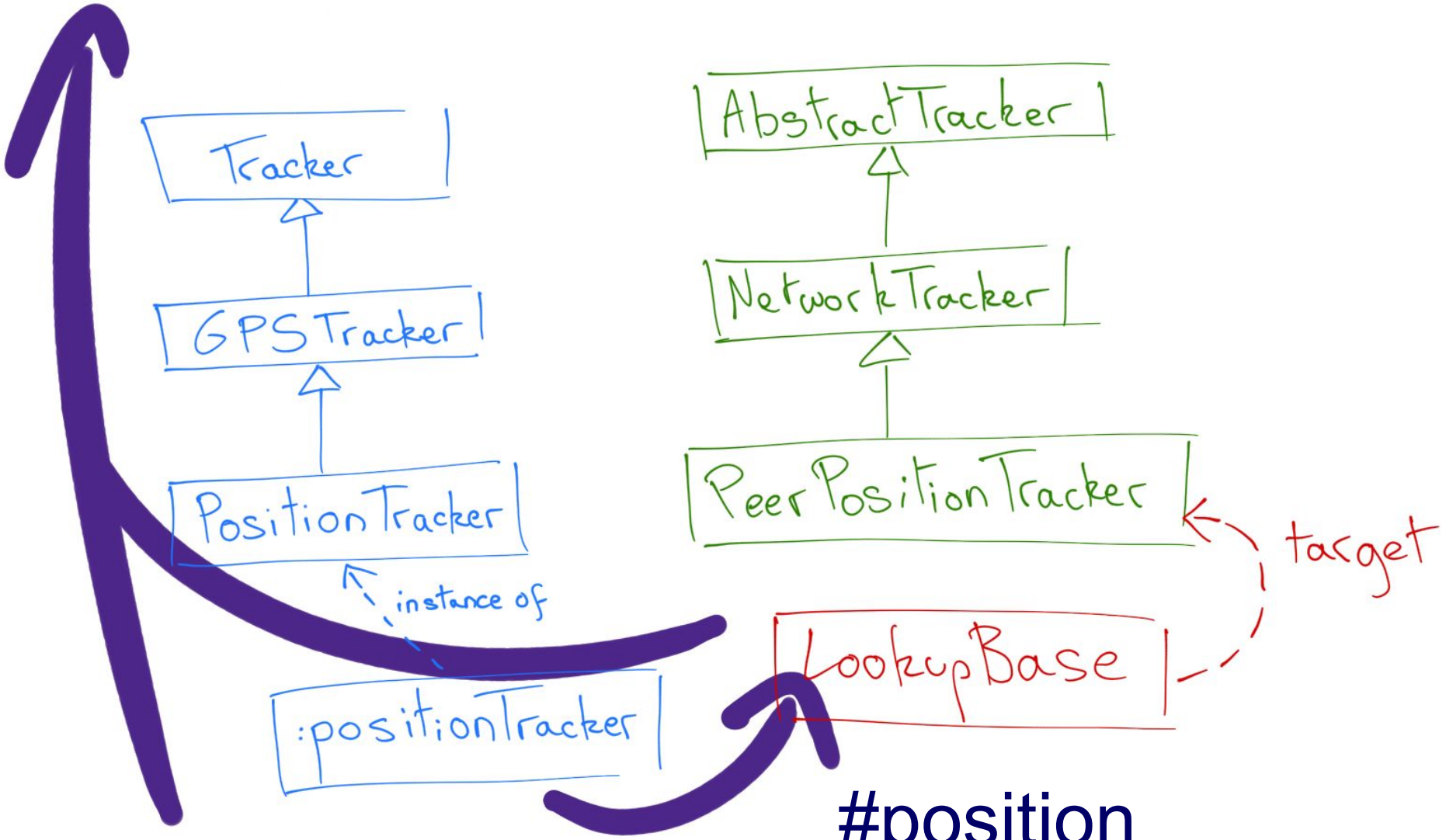
Selection of the behavior to adapt ?

```
lub := LookupBase
        named: 'PeerTrackerLookupBase'
        targetClass: PeerPositionTracker
        with: #(#pinPoint).

positionTracker lookupBase: lub.

positionTracker pinPoint.
positionTracker position.
```


Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
AbstractTracker
NetworkTracker
PeerPositionTracker
target
LookupBase


Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
AbstractTracker
NetworkTracker
PeerPositionTracker
LookupBase
target

#pinPoint

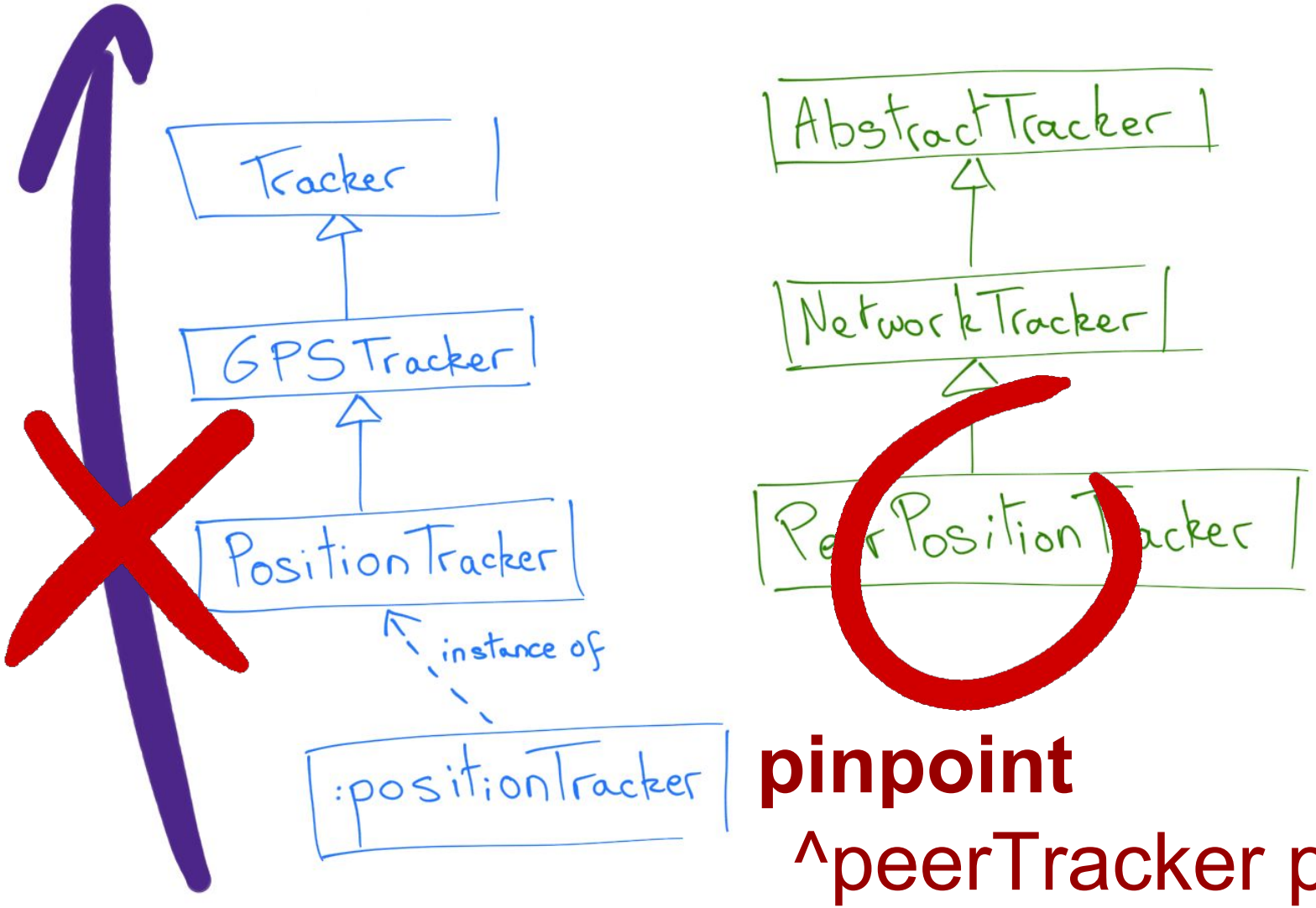

Tracker
GPSTracker
PositionTracker
instance of
:positionTracker
AbstractTracker
NetworkTracker
PeerPositionTracker
target
LookupBase
#position

Binding an object to a meta object that controls the lookup

- Instance based adaptation
- Adapted object preserve their identity
- Free adaptation strategy
- Easy and controlled behavior selection

Issues ?

States consistency with an adaptation ?



pinpoint

^peerTracker pinPoint

Validation and consistency of the adapted behavior ?

Experiments on a physical device ?

Lub
A language for Dynamic Context Oriented Programming

Steven Costiou
Mickaël Kerboeuf, Glenn Cavarlé, Alain Plantec
UMR CNRS 6285, Lab-STICC/MOCS
Université de Bretagne Occidentale