

---

# Managing Areas of Interest In Command and Control Information Systems

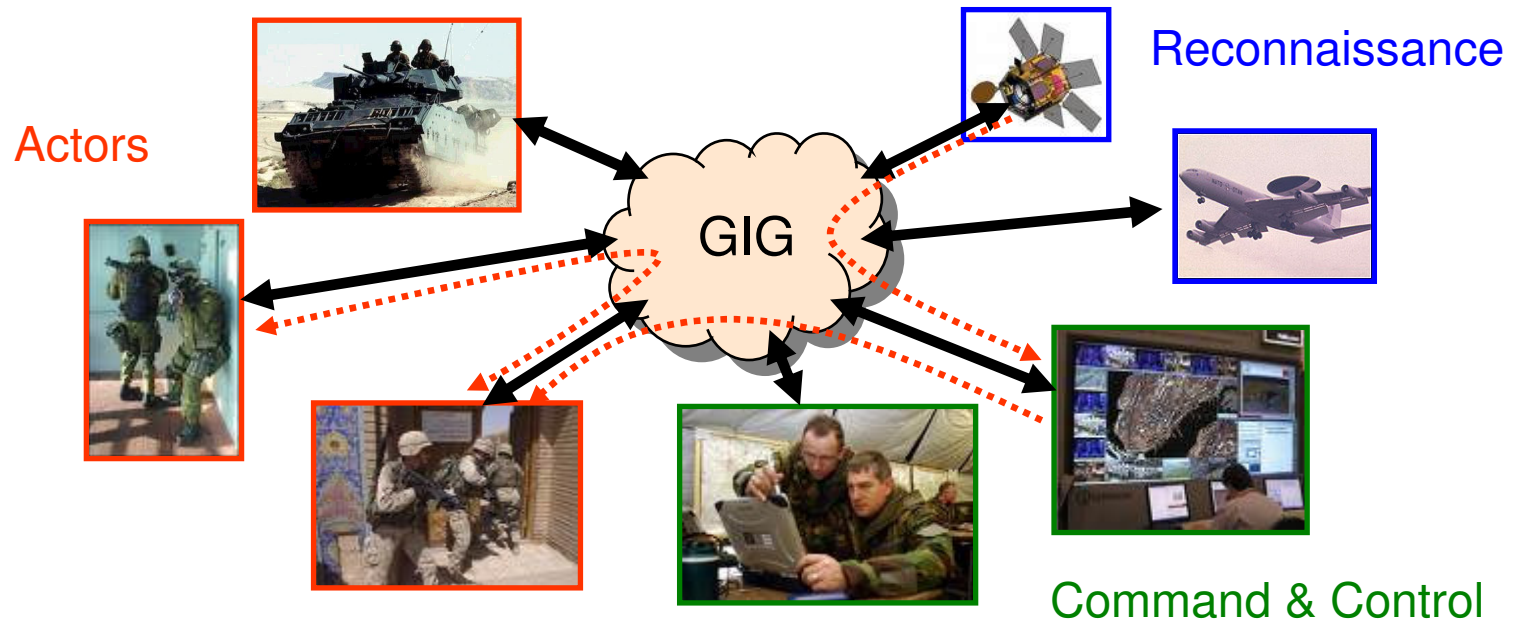
*12th ICCRTS 2007, Newport, RI*

Thomas Nitsche

FGAN/FKIE

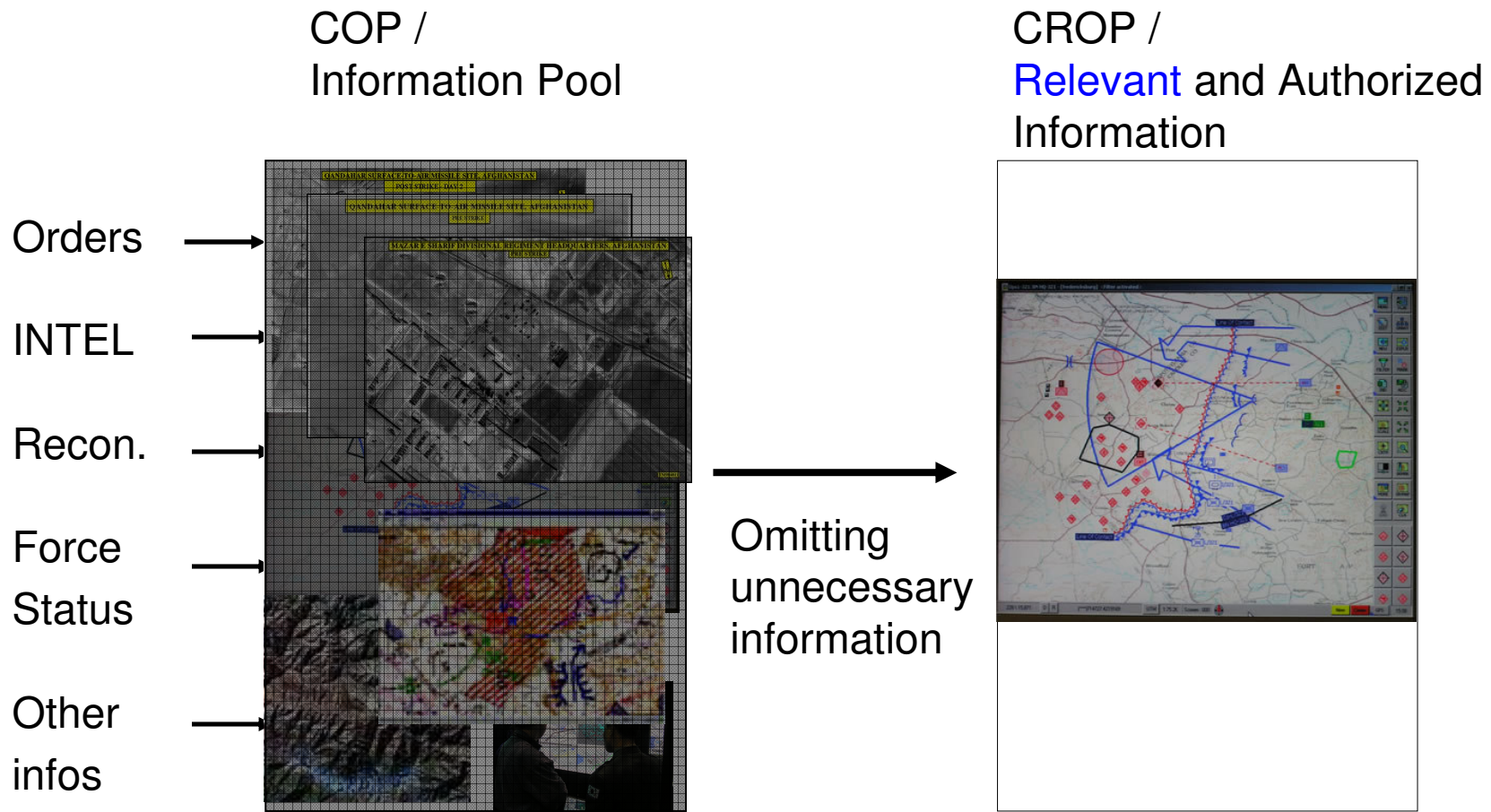
[nitsche@fgan.de](mailto:nitsche@fgan.de)

Combining all nodes in the system allows global information sharing



Problem: Potential information overload (too much data)  
→ Find the **relevant information**

**Filtering** (mission) **relevant information** – the right amount of information (but not more)



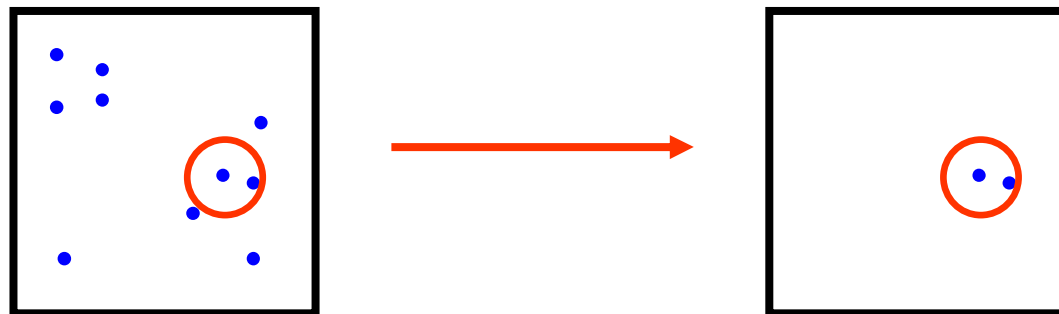
- Area of concern to the commander
- Concentrate on geographically based areas here
- Generally:
  - area of responsibility / operation of corresponding military unit
  - + some surrounding area
- Monitor behaviour of
  - neighbouring forces, as well as
  - status of spatial objects (e.g., streets, bridges)

# Areas of Interest (2)

- AOI: Geographical area around own position
- E.q.: all geo-referenced objects in n-km radius



- More complex areas possible as well
- Here: do not explicitly distinguish blue/red force or other spatial objects
- → generalization: geo-referenced objects with status & capability attributes



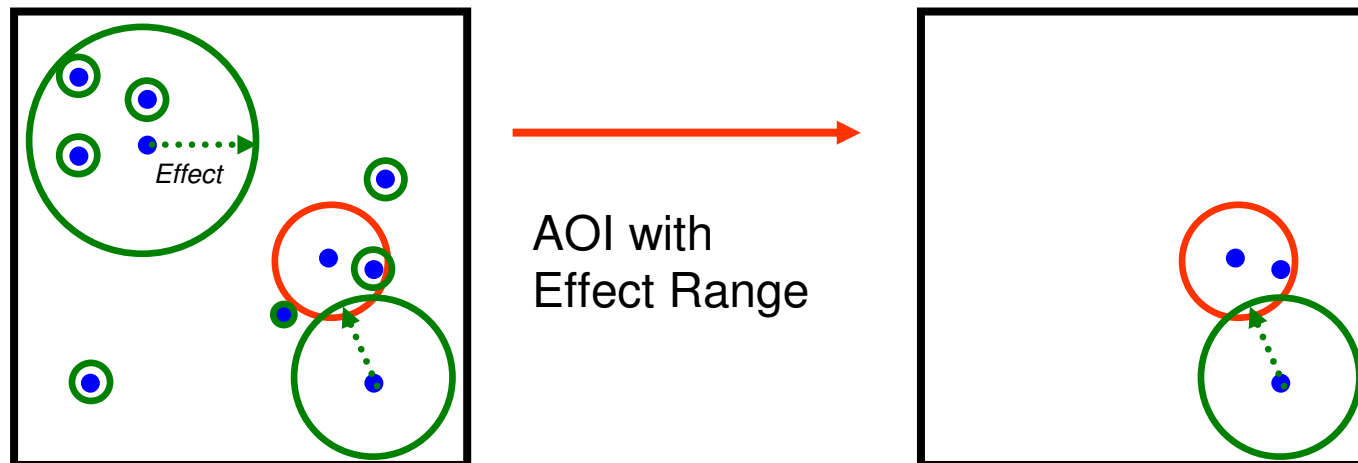
- Do not use AOI as (fixed) surrounding of specific size, but **extend AOI** by
  1. Potential impact of other objects (units) on us, based on their **effect range**
  2. Their potential impact in the **future** based on planned activities

# Extended AOIs (2): Incorporate the Effect Range

7

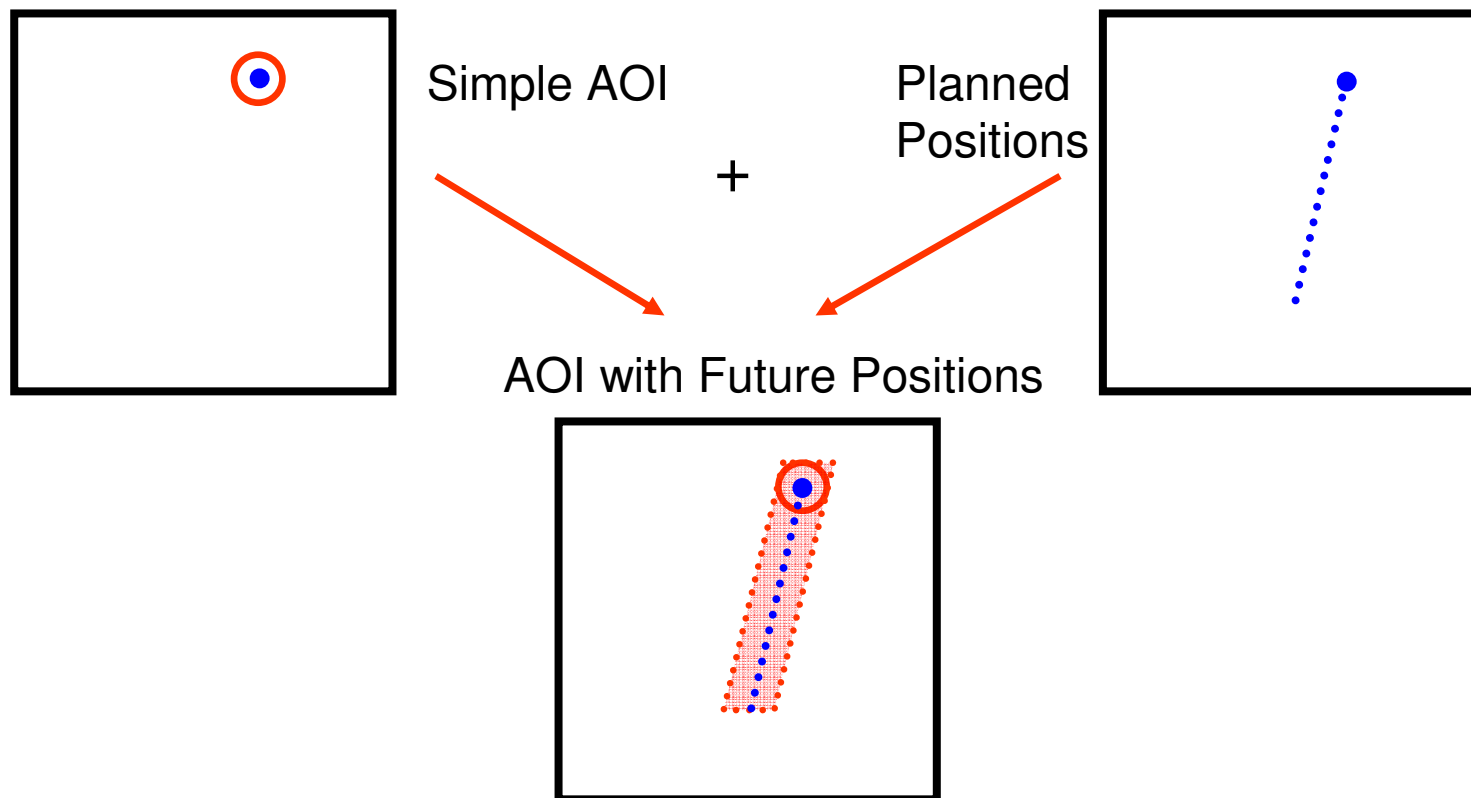
Consider **effect range** of objects (thread or support range), i.e.

- Weapon range
  - Speed, direction and range of movements
- Examples:
    - Artillery: large effect range → remote impact
    - Infantry: local effect



# Extended AOIs (2): Incorporate the Future

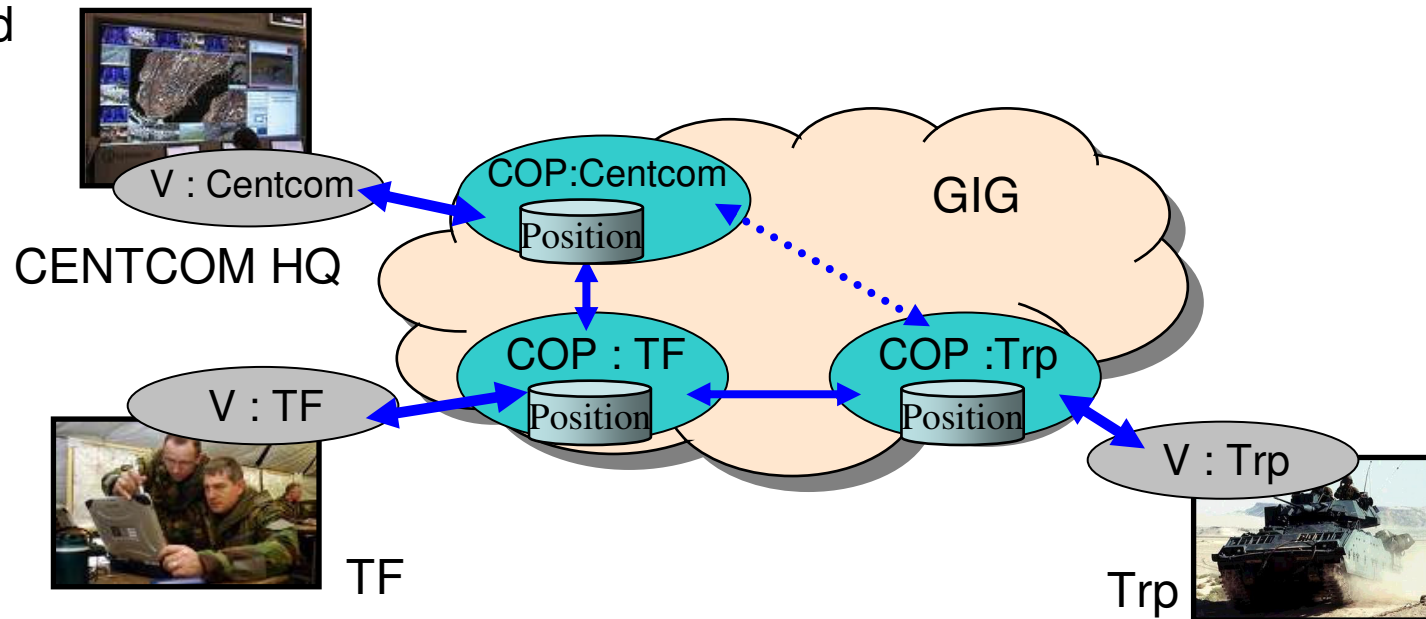
- According planned orders: compute expected positions in **future**
  - Example: flight path or patrol route



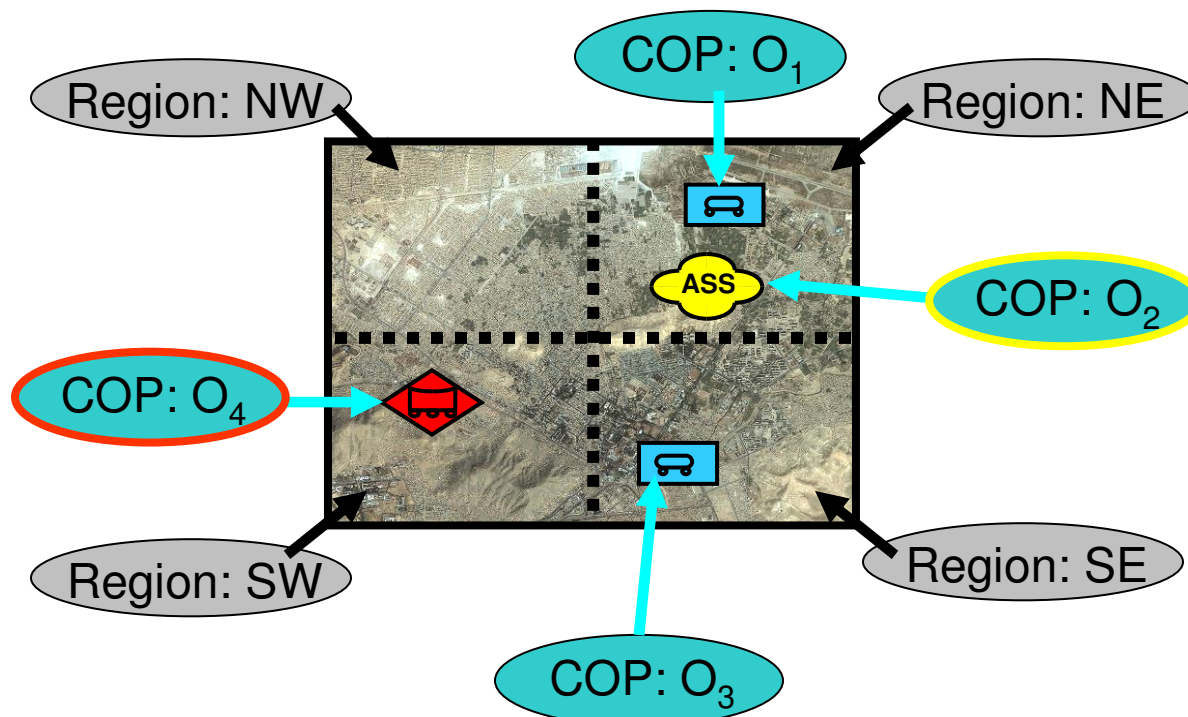


- Simple Approach:
  - Check positions of *all N* objects (units etc.) with GIG
  - Filter objects within AOI
  - complexity:  $O(N)$  for AOI of only one commander
  - Complete C2IS:
    - $O(N^2)$  (naively)
    - $O(N \log N)$  (synchronized all-to-all, e.g. Barnes-Hut Force-Calc. Alg.)
- Other problem: dynamic movements of objects
  - AOI soon outdated, requires re-computation in regular intervals
- Better:
  - Pre-filter (far) remote objects
  - Only consider only consider objects in “near surrounding”
  - → *Region Services*

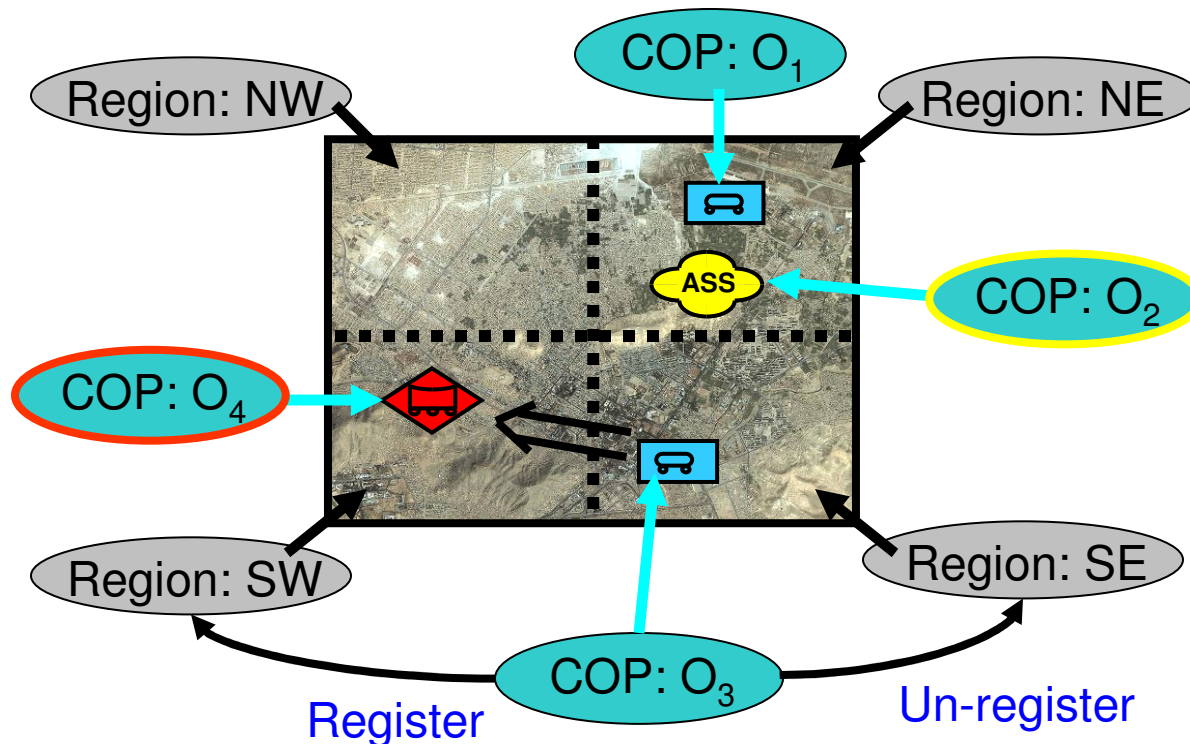
- Functionality provided as **services**, e.g.
  - User-specific **COP-service** instances  
(position, status + operational picture of specific unit)
  - User-specific **visualization-service** instances
- Service instances can be distributed flexible within GIG, modes:
  - Fully decentralized operation, shared data via publish-subscribe
  - Central database, (virtual) service instances on central server
  - mixed



- Region services integrate directly into service-oriented C2IS architecture
- Idea: distribute world/area into **regions**
- **Region service** instances
  - Handle one specific region
  - Contain all objects within that region

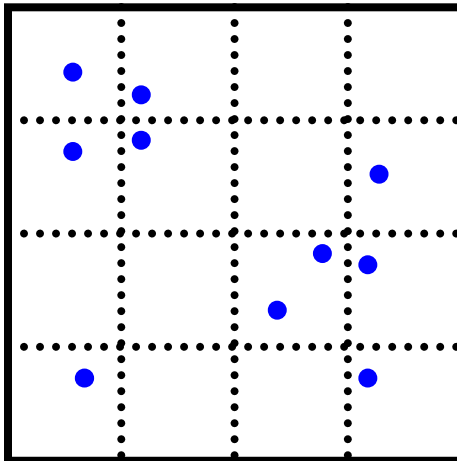


- Dynamic movements of objects
  - Objects (un-)register themselves at region service instance, if they enter (leave) a region
  - Example:  $O_3$  attacks (and moves towards)  $O_4 \rightarrow$  change region

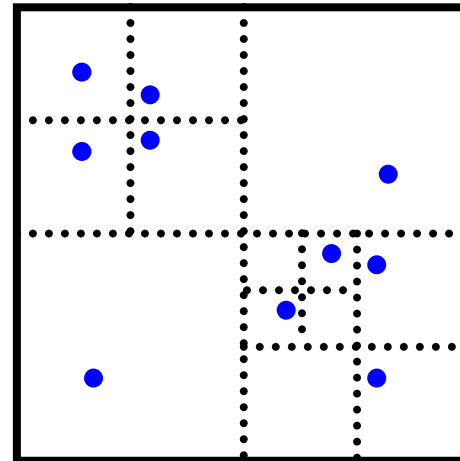


- Region distribution
  - a) Regular regions
  - b) Hierarchical regions  
(adopting regions to
  - c) Use borders of areas of operation

(a) Regular Regions



(b) Hierarchical Regions (Quad-Tree)



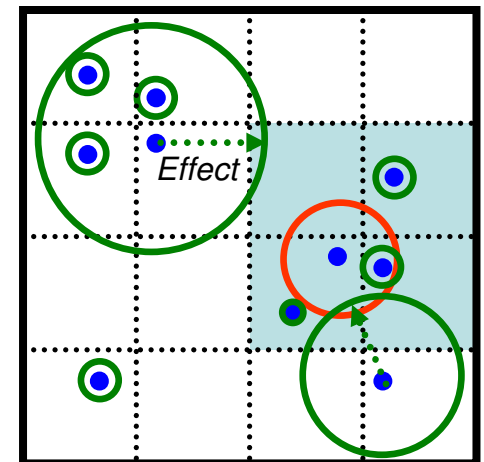
# AOI Computation using Region Services with Effect Range 14

- General idea:
  1. Initialization: Object registration
  2. For each user:

Determine relevant regions and objects within that regions

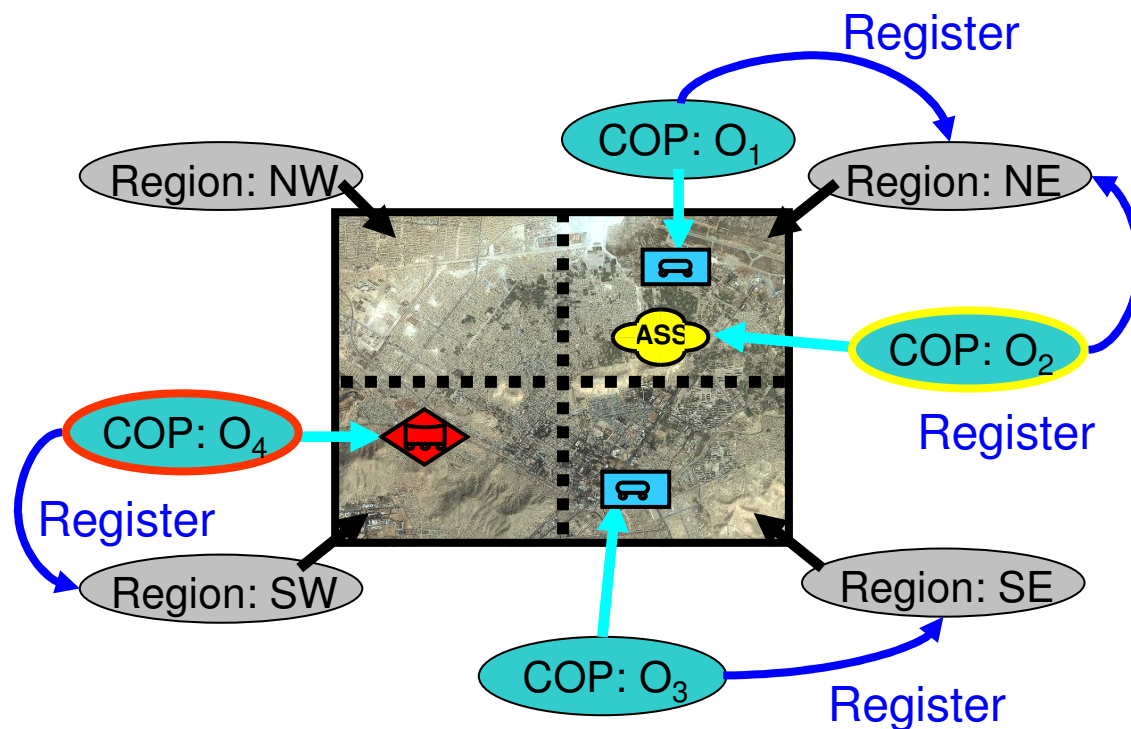
1. (Local) computation of AOI

- Detailed description follows ...

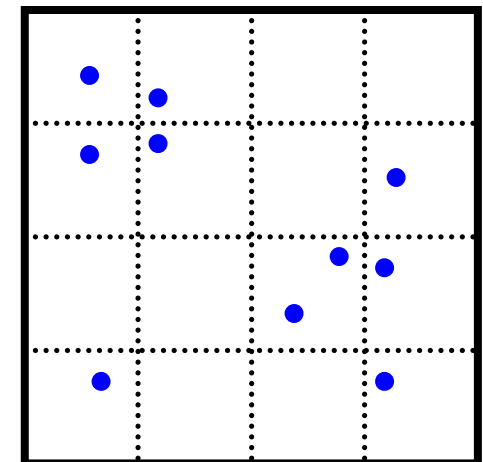


- Phase 1 (Initialization):  
 Register objects at corresponding region services

Example

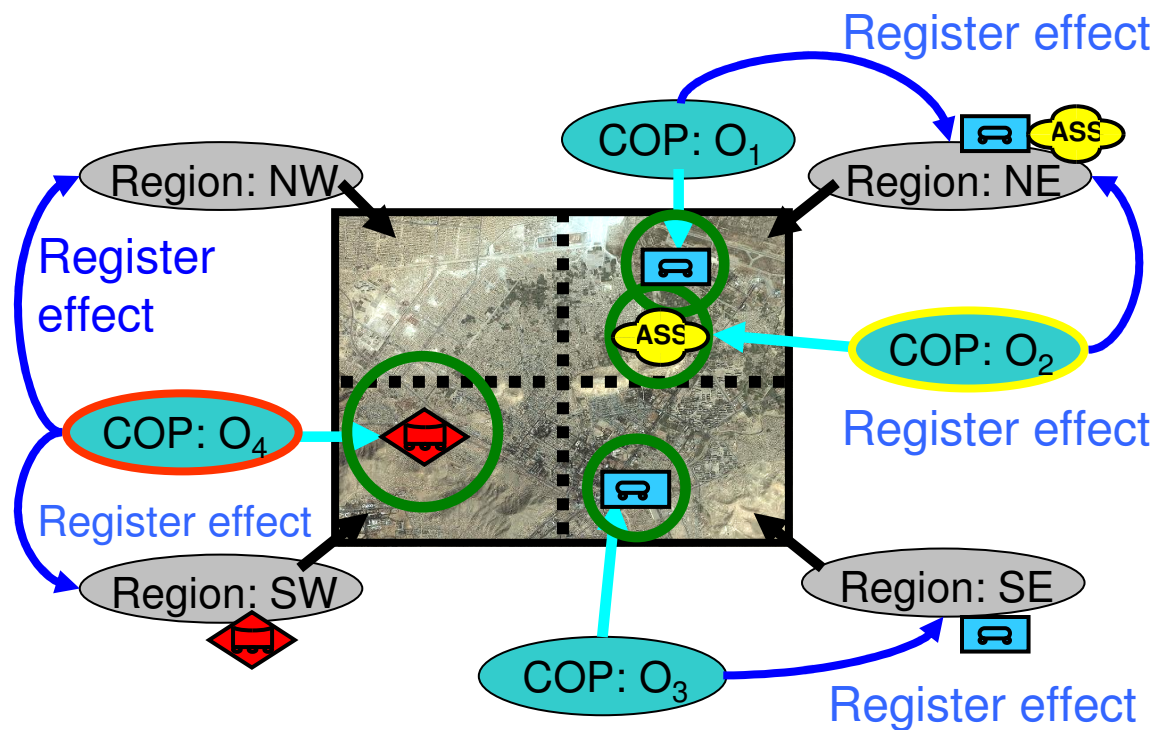


Abstract view

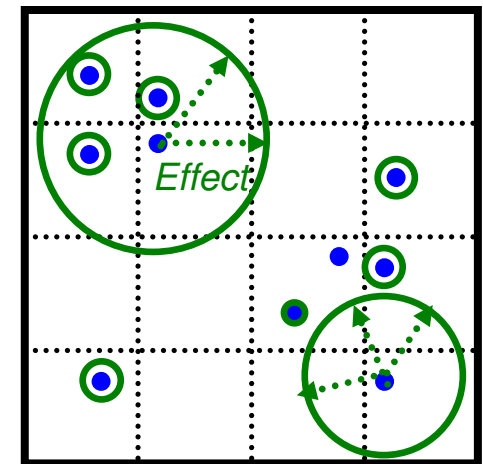


- Phase 2 (Initialization Cont.):  
Register effect range of objects at region services

Example

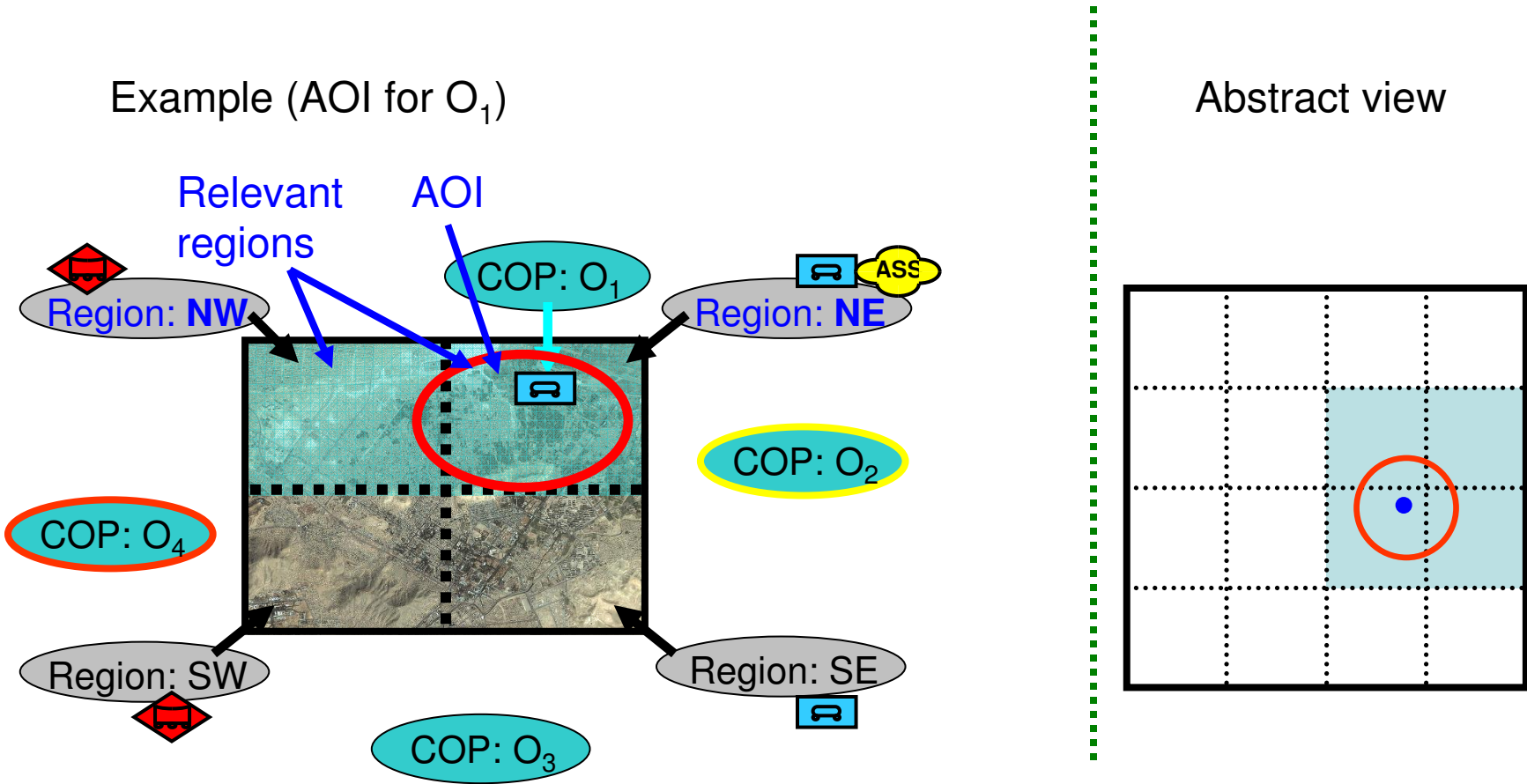


Abstract view



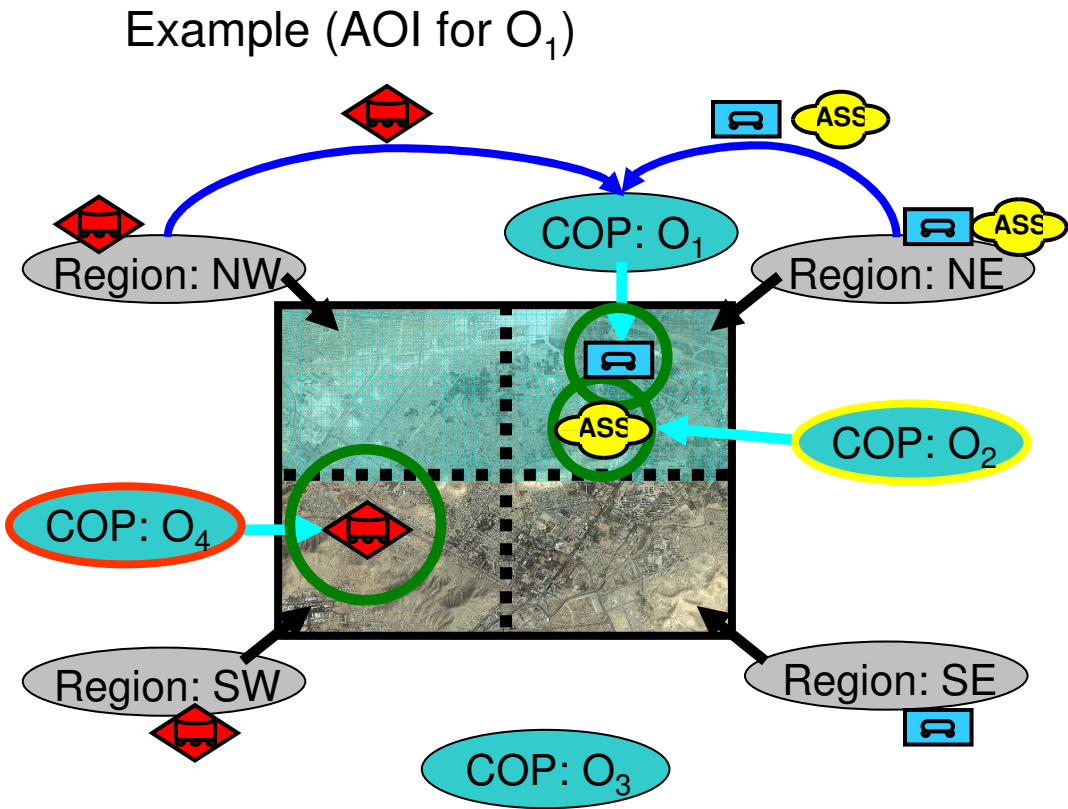


- Phase 3 (Start Object Determination):  
Determine relevant regions for AOI (for each user)

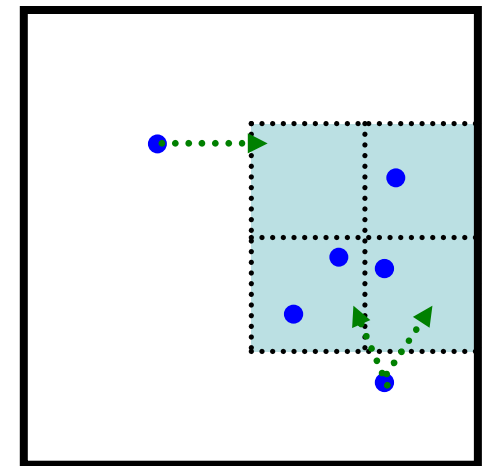


- Phase 4:

Get object sets from relevant region services

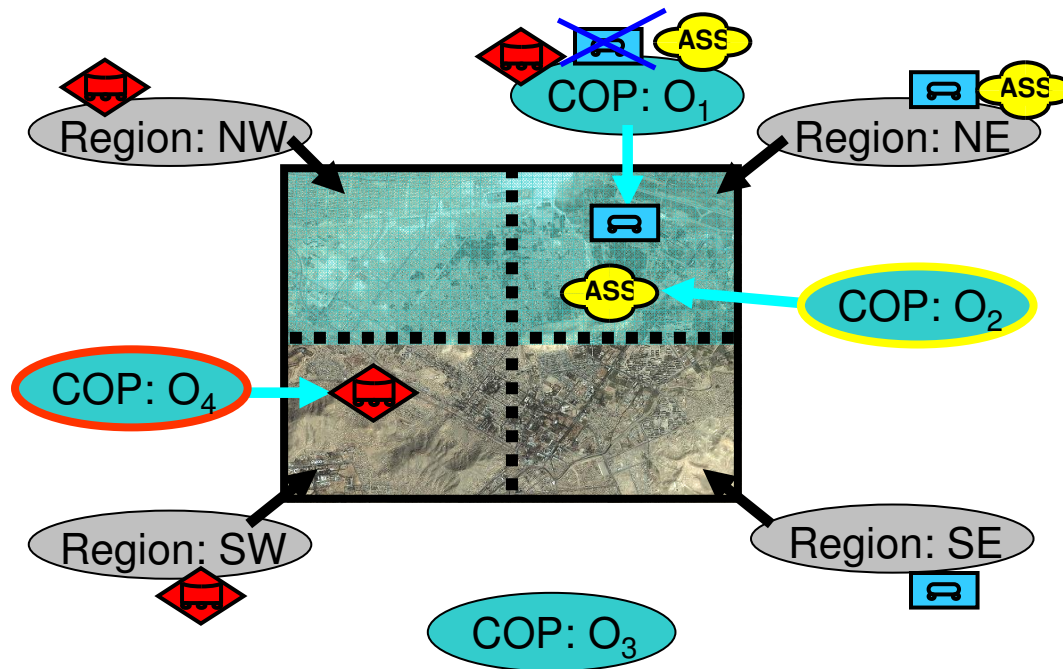


Abstract view

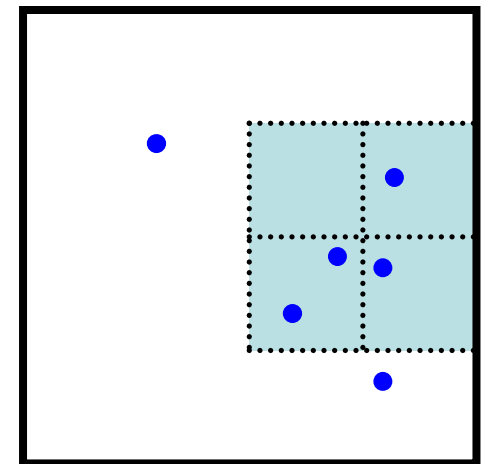


- Phase 5:  
Merge object sets

Example (AOI for  $O_1$ )

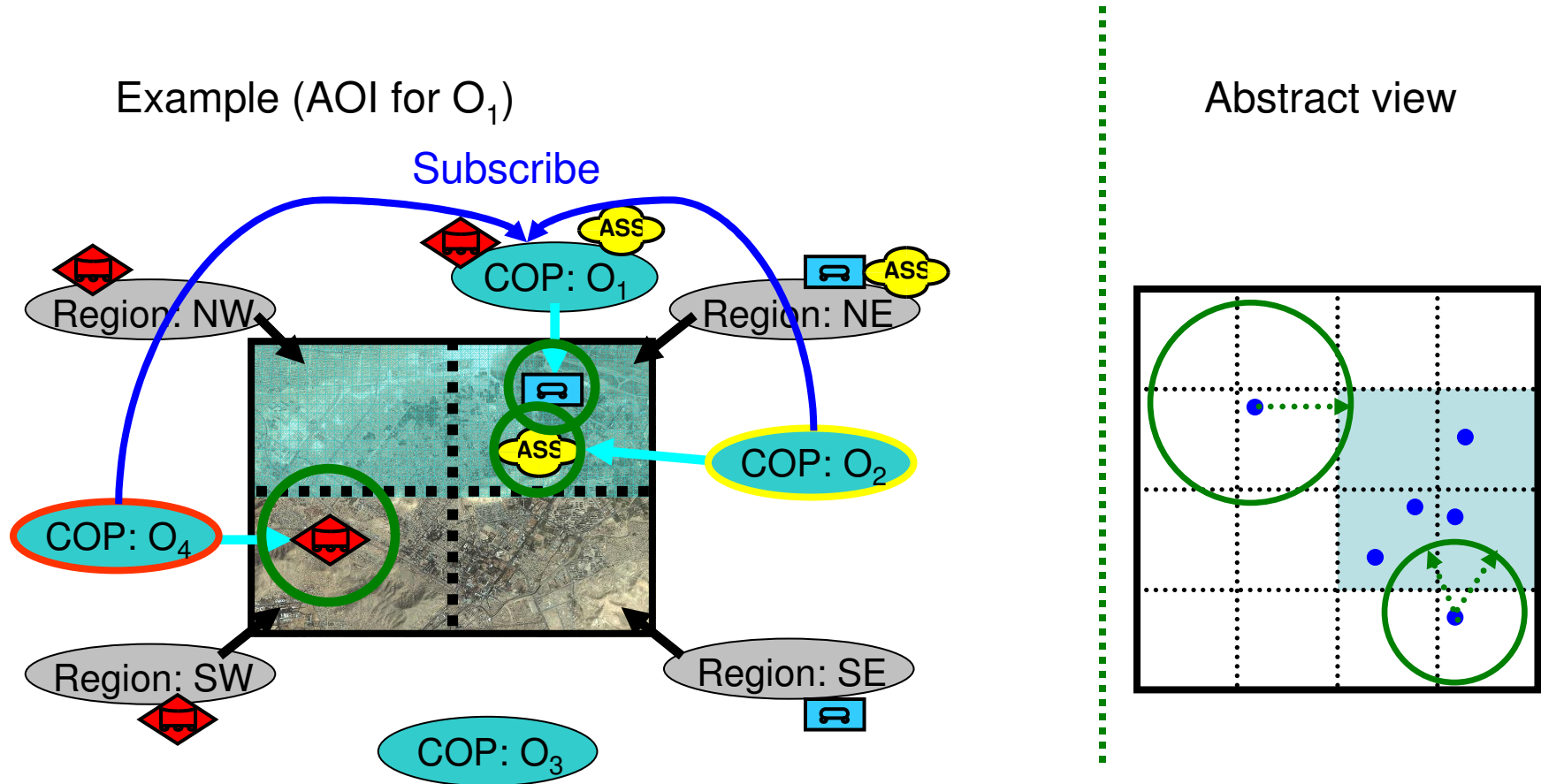


Abstract view



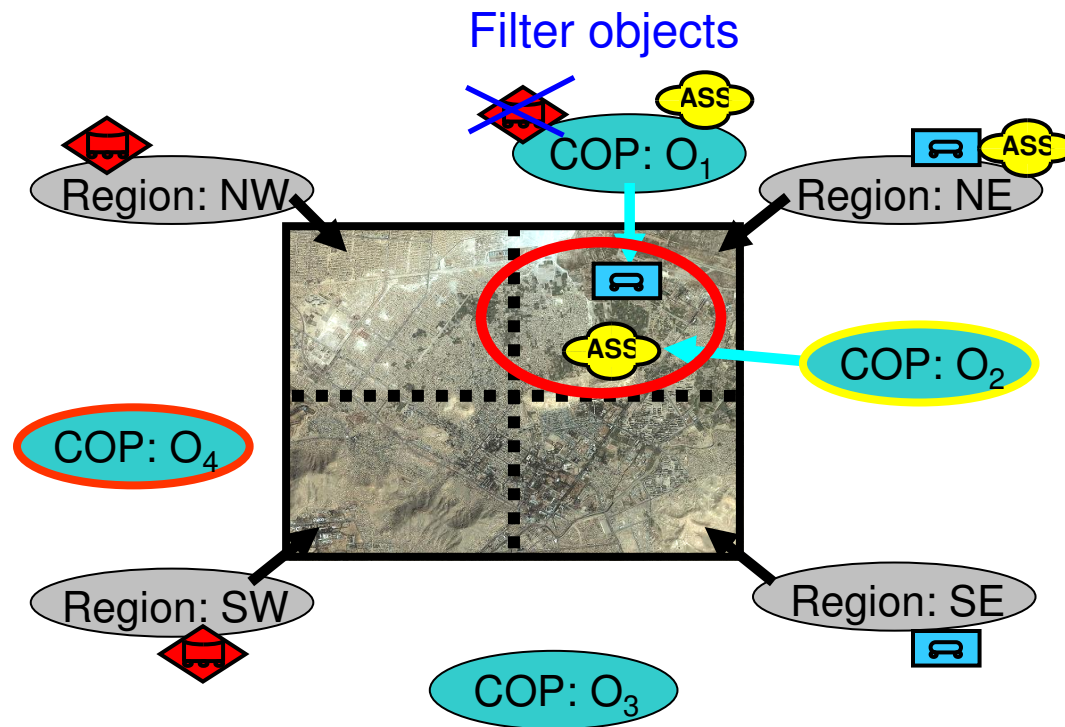
- Phase 6:

Subscribe to objects: Read object positions and effect range

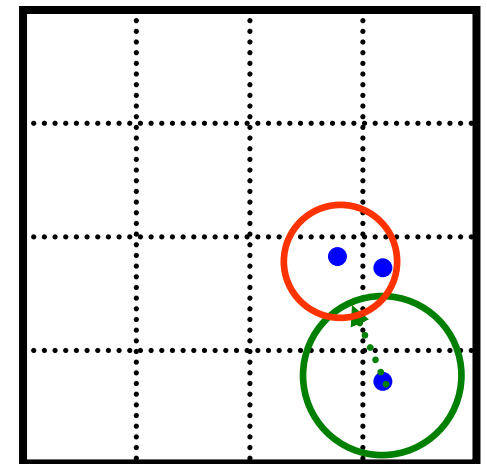


- Phase 7 (Local):  
Compute AOI: Determine objects with AOI

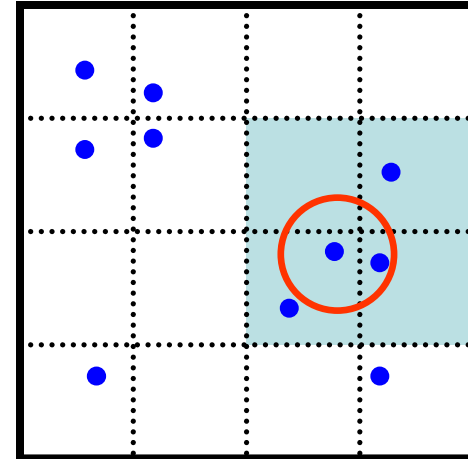
Example (AOI for  $O_1$ )



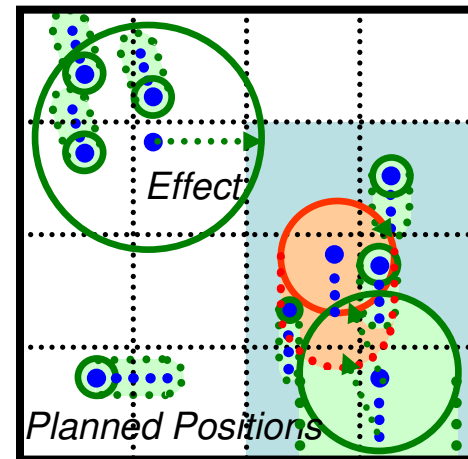
Abstract view



- AOI without effect range:  
→ Omit phases 2 and 5 (merge)



- Incorporating Future Positions:  
Similarly,  
but more complex shape of AOI



- Areas of interest allow information filtering within GIG
- Introduced concept of region services
  - Handle all objects within specific geographic area
  - Allow efficient observation of AOI
  - Provided efficient AOI computation algorithm
  - Allows AOI with effect range (and future) of objects
- Future Work
  - Evaluate different location management schemes
  - Variation of region distributions to better handle in-homogenous distributions of objects

- Areas of interest allow information filtering within GIG

- Introduced concept of region services

# Questions?

- Handle a objects with respect to their appearance
- Allow efficient observation of AOI
- Provided efficient AOI computation algorithm
- Allows AOI with effect range (and future) of objects

- Future Work

**Thomas Nitsche**

- Evaluate different location management schemes
- Variation of region distributions to better handle in-homogenous distributions of objects

**FGAN/FKIE**

**nitsche@fgan.de**