Engineering the aggregate

Danilo Pianini
danilo.pianini@unibo.it

Alma Mater Studiorum—Università di Bologna

Software Engineering for Intelligent and Autonomous Systems
2017-08-25 – Dagstuhl, Germany, EU

Outline

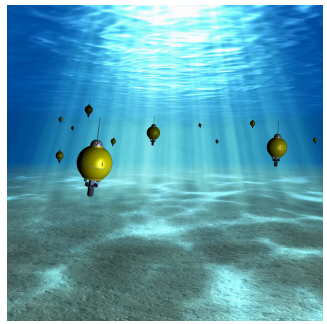
The world is becoming a crowded and complex place

Future and emerging Internet-of-things are witnessing:

- devices increasingly wearable, mobile, embedded, flying...
- increasing availability of heterogeneous wireless connectivity
- increasing availability of computational resources (device/edge/cloud)
- increasing production and analysis of data, everywhere, anytime

$\Rightarrow$ business, security, and privacy concerns will be drivers as well



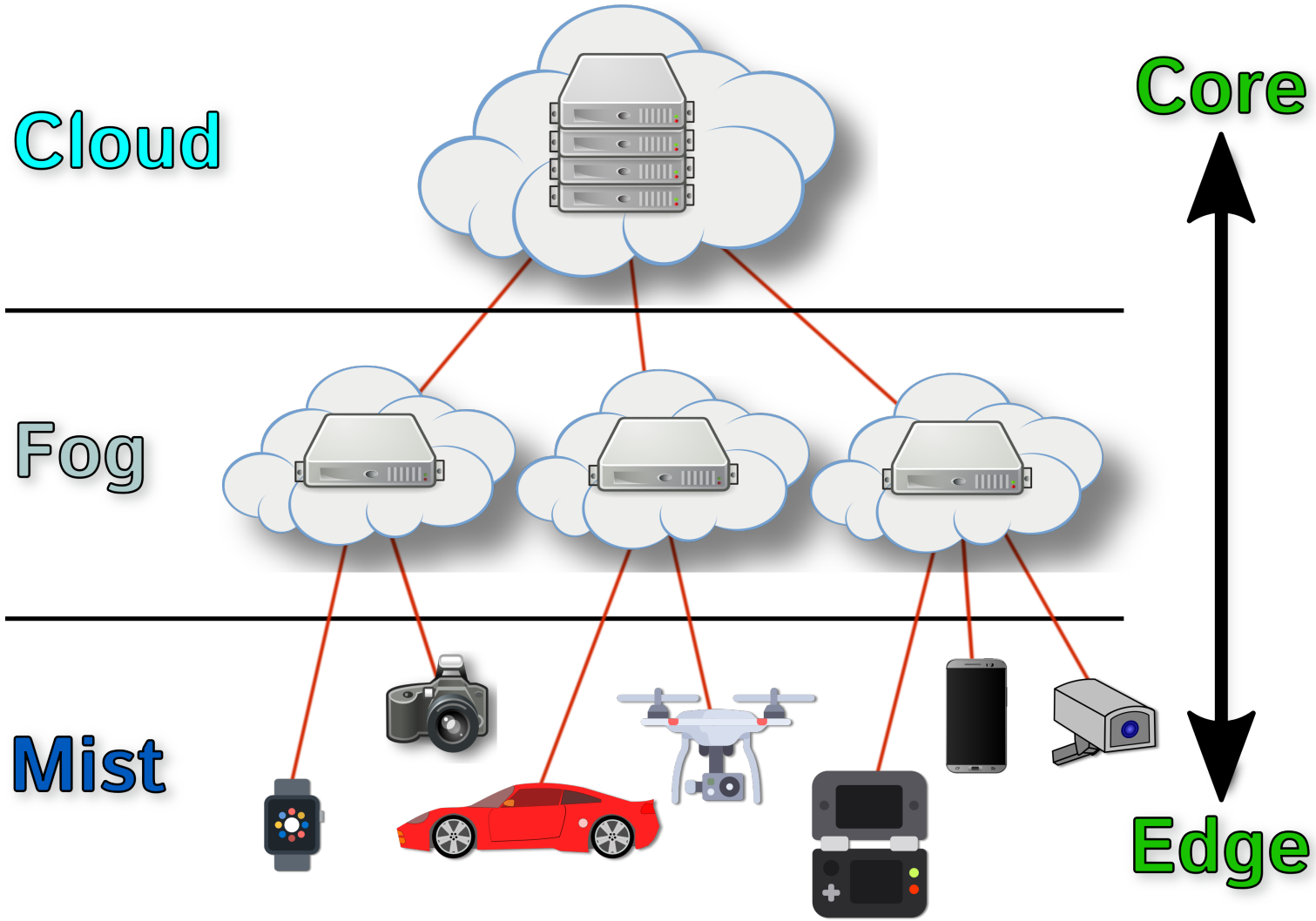



Outline

Programming models for mobile and IoT applications

- Client side
  - Single-device program: objects, functions, actors, futures, tasks, activities
  - Local intelligence
- Client-server communication
  - SOA (Http/Rest), MoM (RabbitMQ)
- Server side
  - Storage: OO, relational, NoSQL — in memory or on-disk
  - Coordination (orchestration, mediation, rules enactment)
  - Situation recognition (online/offline, mining, business intelligence, stream processing)
- Scalability server-side calls for cloud-ification
  - Not really orthogonal to the whole programming model
  - It often dramatically affects system design

Heterogeneity of deployment contexts



Implications

Where does programming effort end up into?

Programs of clients and servers highly depend on:

- Platform and API
- Communication technology
- Number, type, and dislocation of devices

$\Rightarrow$ Design and deployments hardly tolerate changes

$\Rightarrow$ IoT systems tend to be rigid, and costly to debug, evolve, and maintain

The technological result

- Few of the opportunities of large-scale IoT are currently exploited
  - Virtually, any computational mechanism (sensing, actuation, processing, storage) could involve spontaneous, adaptive cooperation of large sets of devices
- How many large-scale deployments of **adaptive** IoT systems around?
- Where are the "Collective Adaptive Systems"?

A programming model perspective

What do we lack in large-scale IoT systems?

- The plain old platform-independent programming abstraction
- Delegating to the underlying platform everything but the core logic:
  - non-functional aspects (network performance, battery saving...)
  - deployment issues

The challenge

Let's tackle the worst possible scenario:

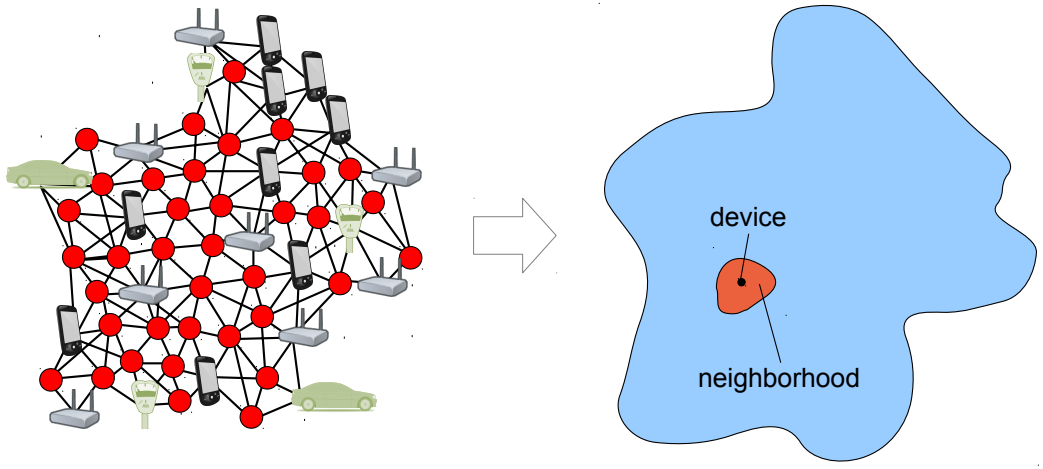
- Countless heterogeneous, moving devices unpredictably situated
- Pervasive sensing and actuation
- Contextual (spatio-temporal) behavior
- Abstract away the infrastructure
  - We can't rely on the cloud or fog
  - But we must be ready to exploit the opportunity they provide

Outline

Manifesto of aggregate computing

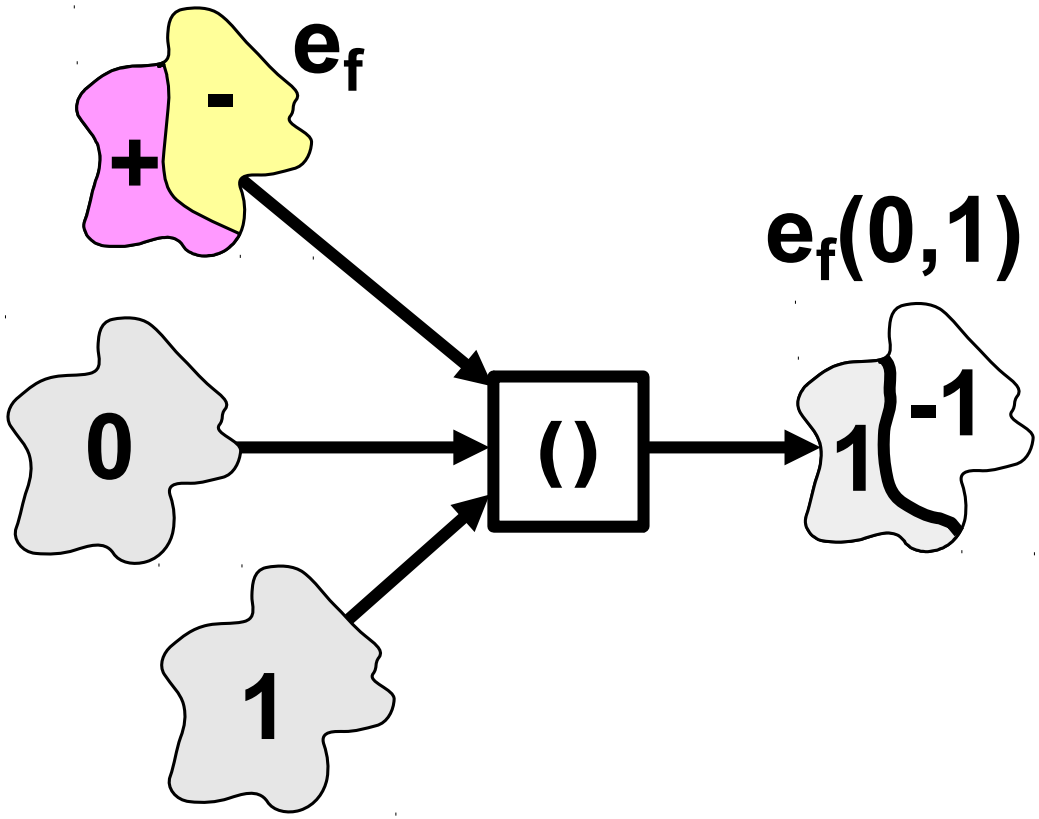
Motto: program the aggregate, not individual devices!

1. The reference computing machine
   $\Rightarrow$ an aggregate of devices as a single entity fading to the actual *space*
2. The reference elaboration process
   $\Rightarrow$ atomic manipulation of a collective data structure
3. The actual networked computation
   $\Rightarrow$ a proximity-based self-org system hidden "under-the-hood"



Computing with fields

Field: a distributed data structure mapping each device to a value



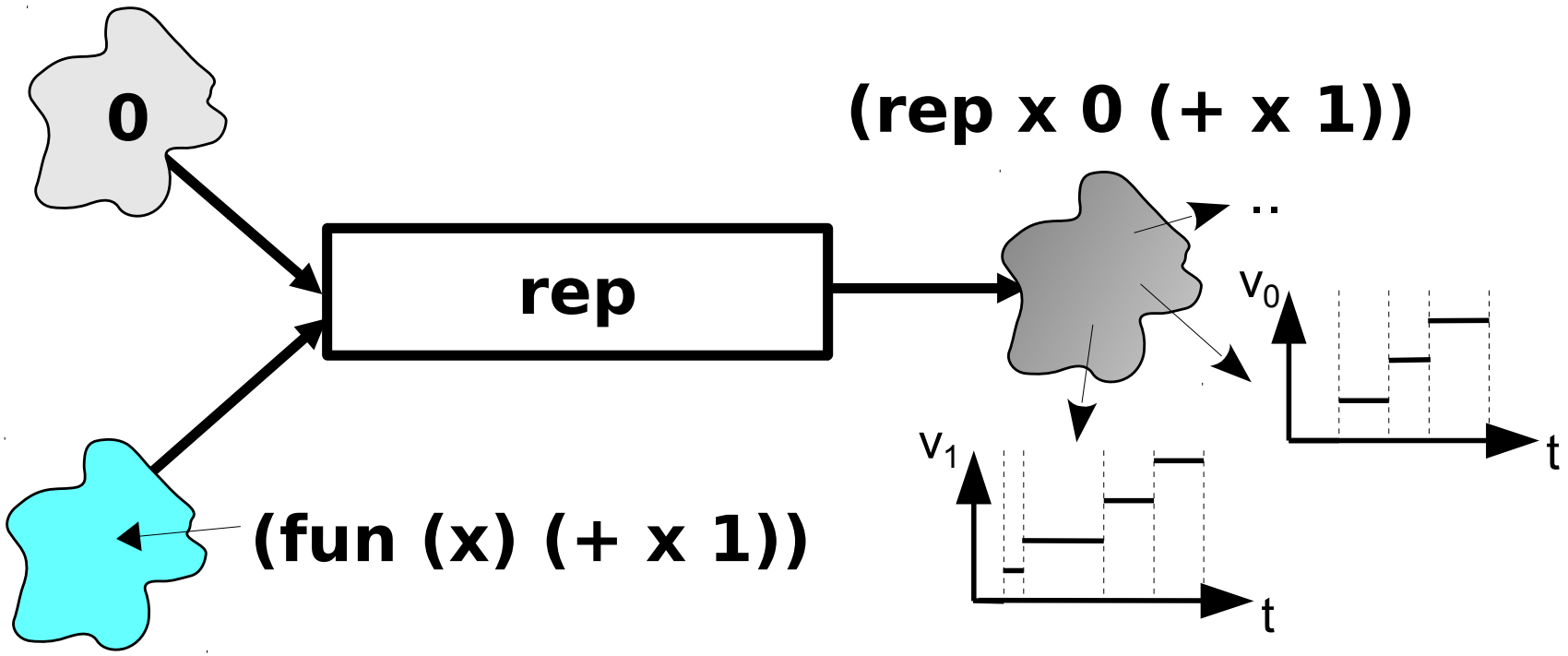
Language based approach

- The approach naturally maps on a new programming paradigm
- A core calculus for such language families exists [DVPB15]
  - Allowing for mathematical proofs of properties
  - Proofs valid for the calculus hold for any practical language based on it
- Aggregate programming languages are proven to be universal...
- ...nevertheless, integration with existing language and libraries is key
  - Don't reinvent the wheel

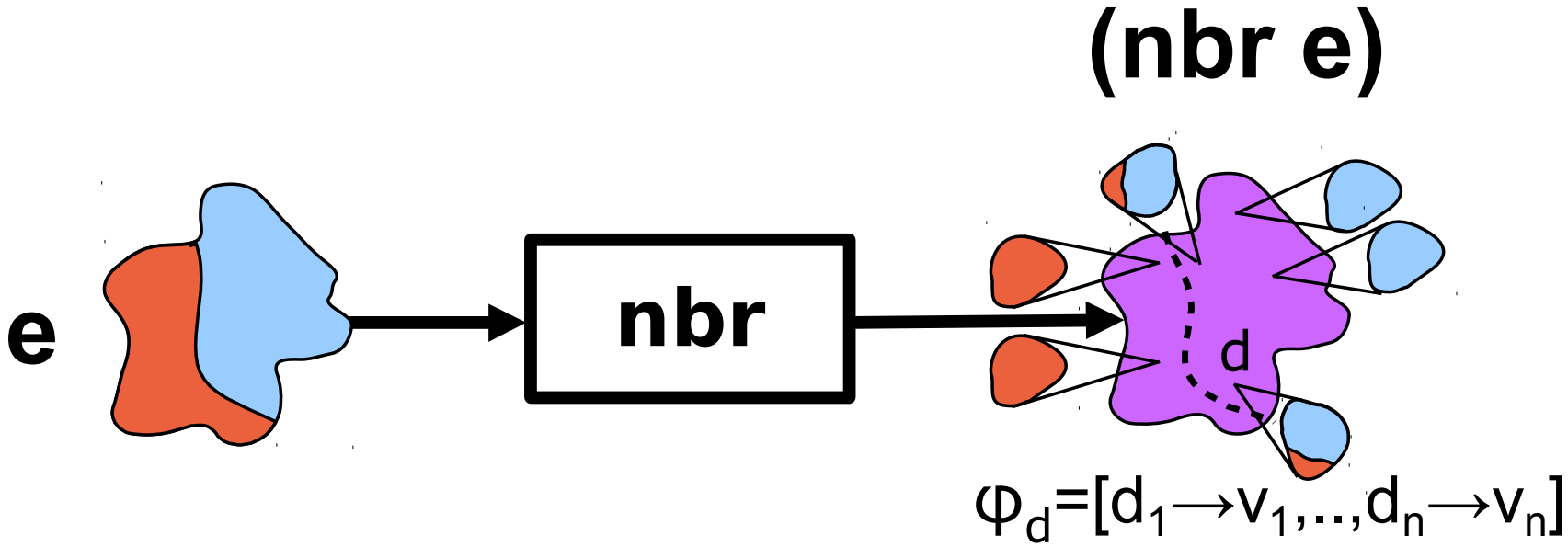
What about a middleware?

- Middlewares have explicit APIs, which we want to avoid
- Nevertheless, a middleware is required for practical languages to work
  - For networking, mostly
- Aggregate programming could be used to easily program some features of novel middlewares as well

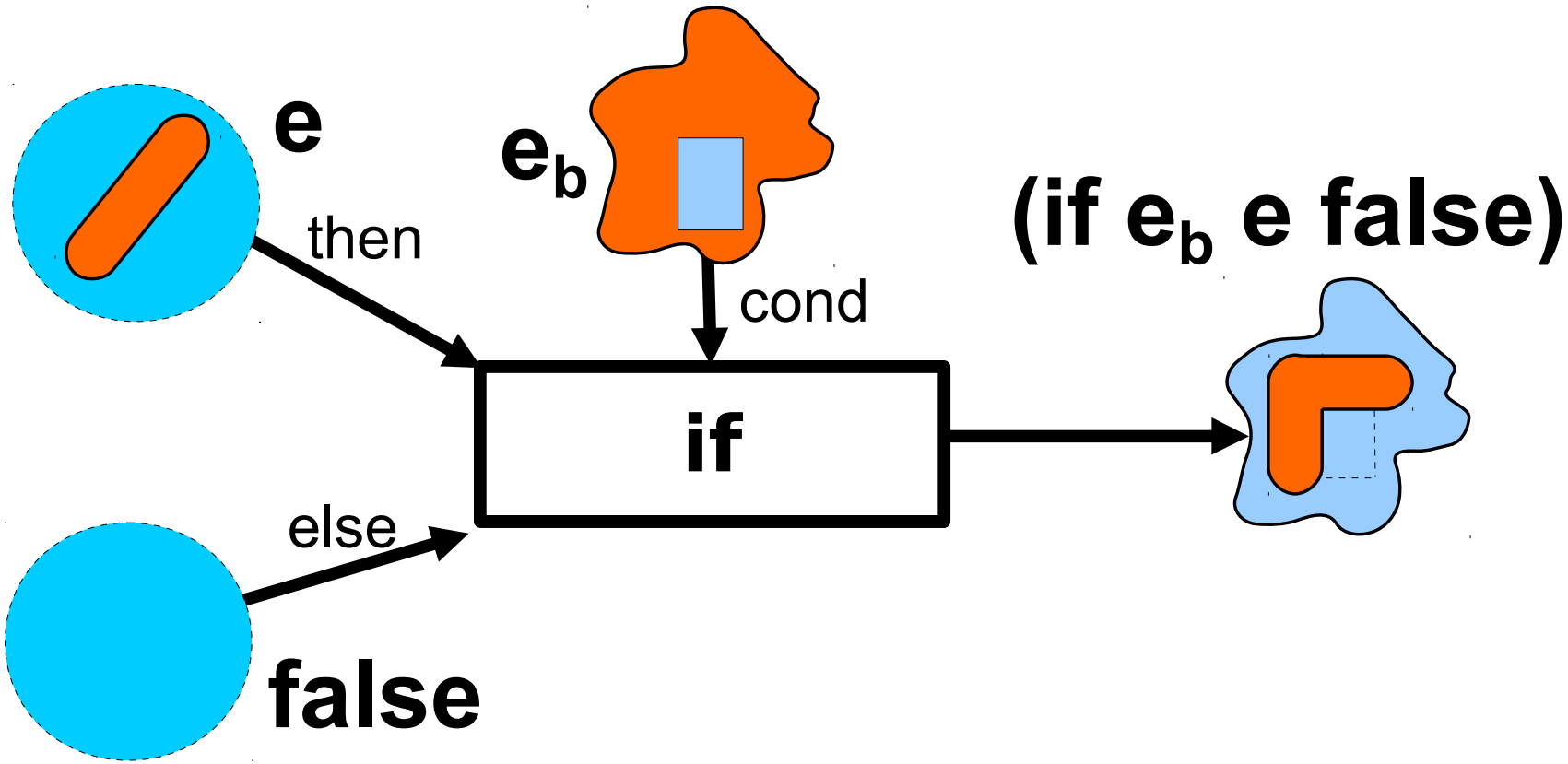
Core language constructs: evolution over time



Core language constructs: gather data from neighborhood

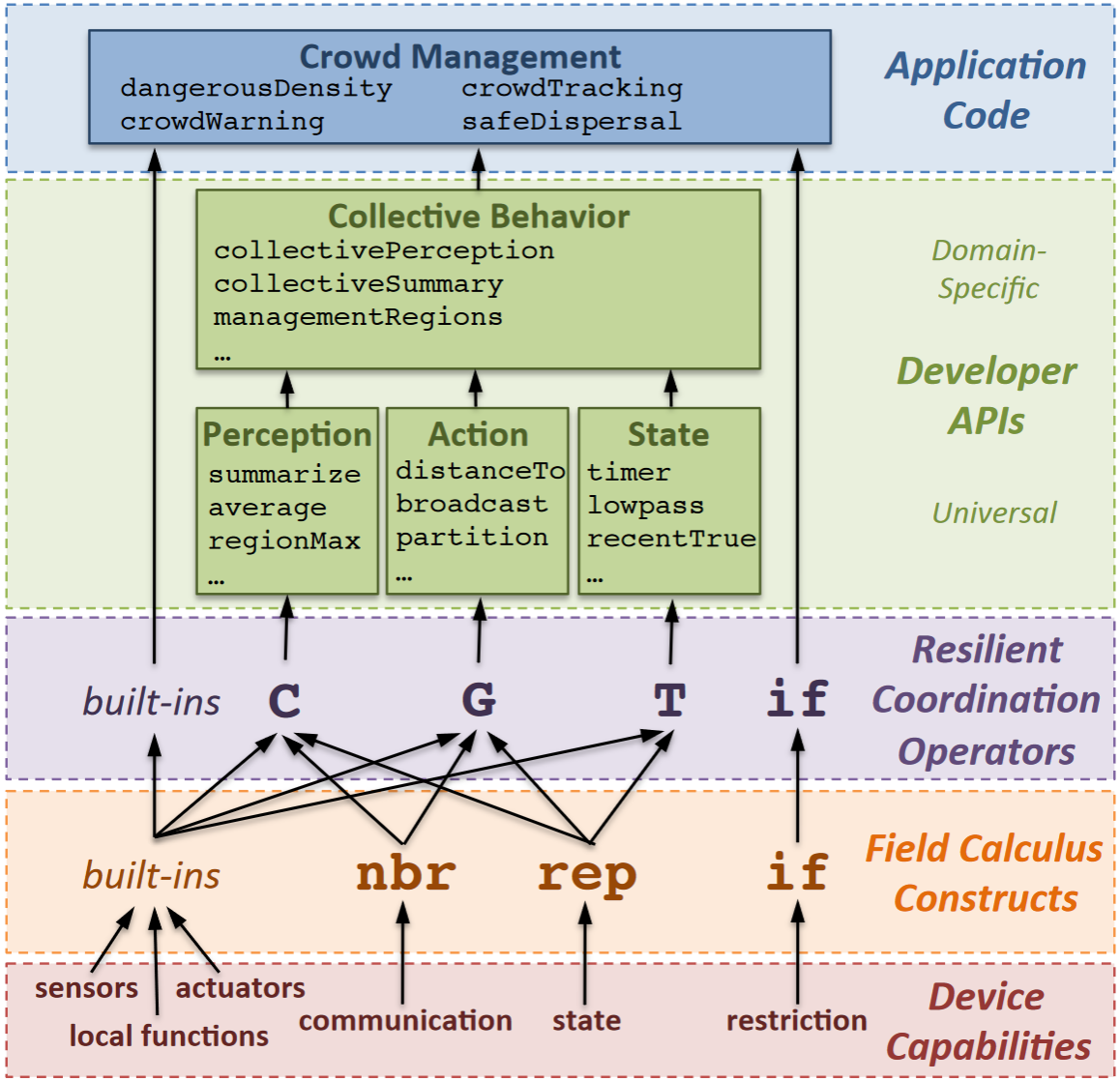


Core language constructs: domain restriction (distributed `if`)



The channel computes in a different domain, but does not get disrupted!

Scalability and compositionality [BV14, VBDP15, FPBV17]



Toolchain I

A set of practical tools are available for engineering the aggregate

Practical languages

- **Protelis**[a] [PVB15, PBV17] – Stand-alone programming language, duck-typed, Java-interoperable (with some limitations), rich library, mature
  - SASO tutorial available to learn it at
- **Scafi** [CV16] – Scala internal DSL, inherits the Scala type system

[a] `www.protelis.org`

Toolchain II

Simulation platforms

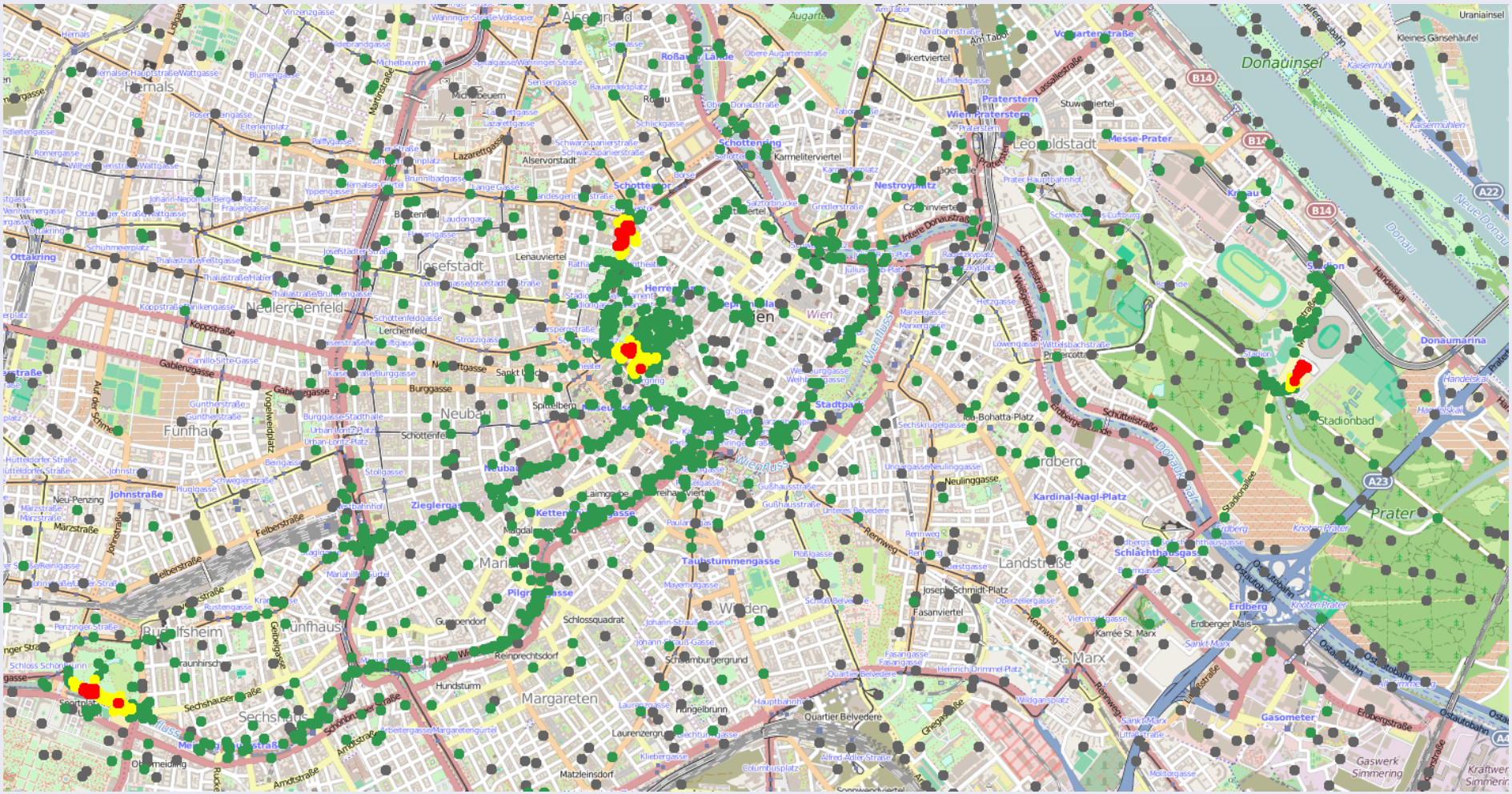
Simulation is paramount: is the only way to test and reproducibly debug a situated self-organizing system.

- **Alchemist**[a] [PMV13, VCP16b] – By far the most largely used simulator when programming in aggregate, supports both Protelis and Scafi, supports (bidimensional) euclidean spaces, floor plans, real world maps (including on-street navigation and GPS traces)
- **NASA World Wind and other simulators**[b] – Protelis can be attached reasonably easily to Java simulation platforms, NASA World Wind has been used in the past for simulating airborne drones [BUL+18].

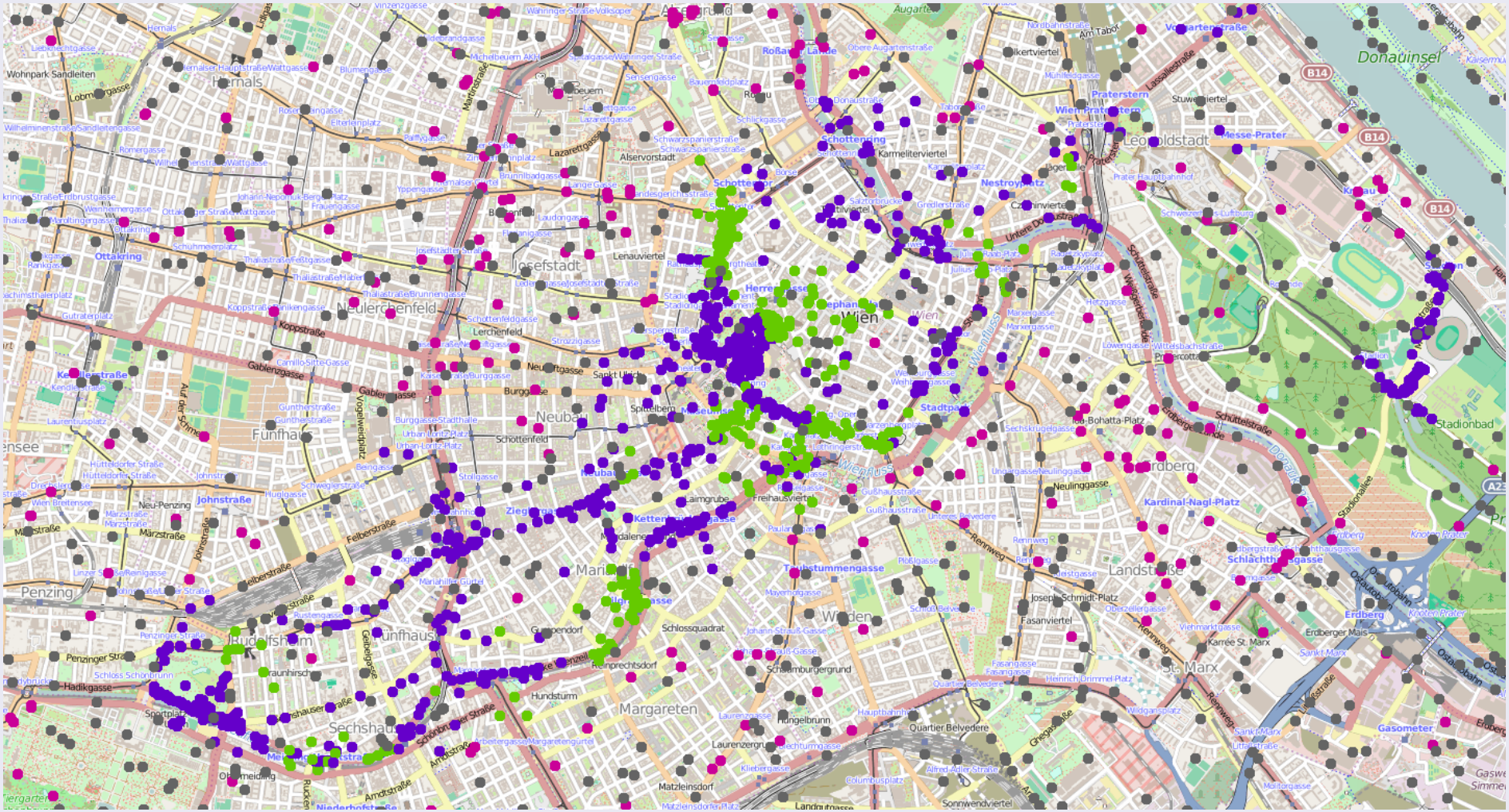
[a] `www.alchemistsimulator.github.io`
[b] `https://github.com/Protelis/Protelis-Demo-Visualized`

Outline

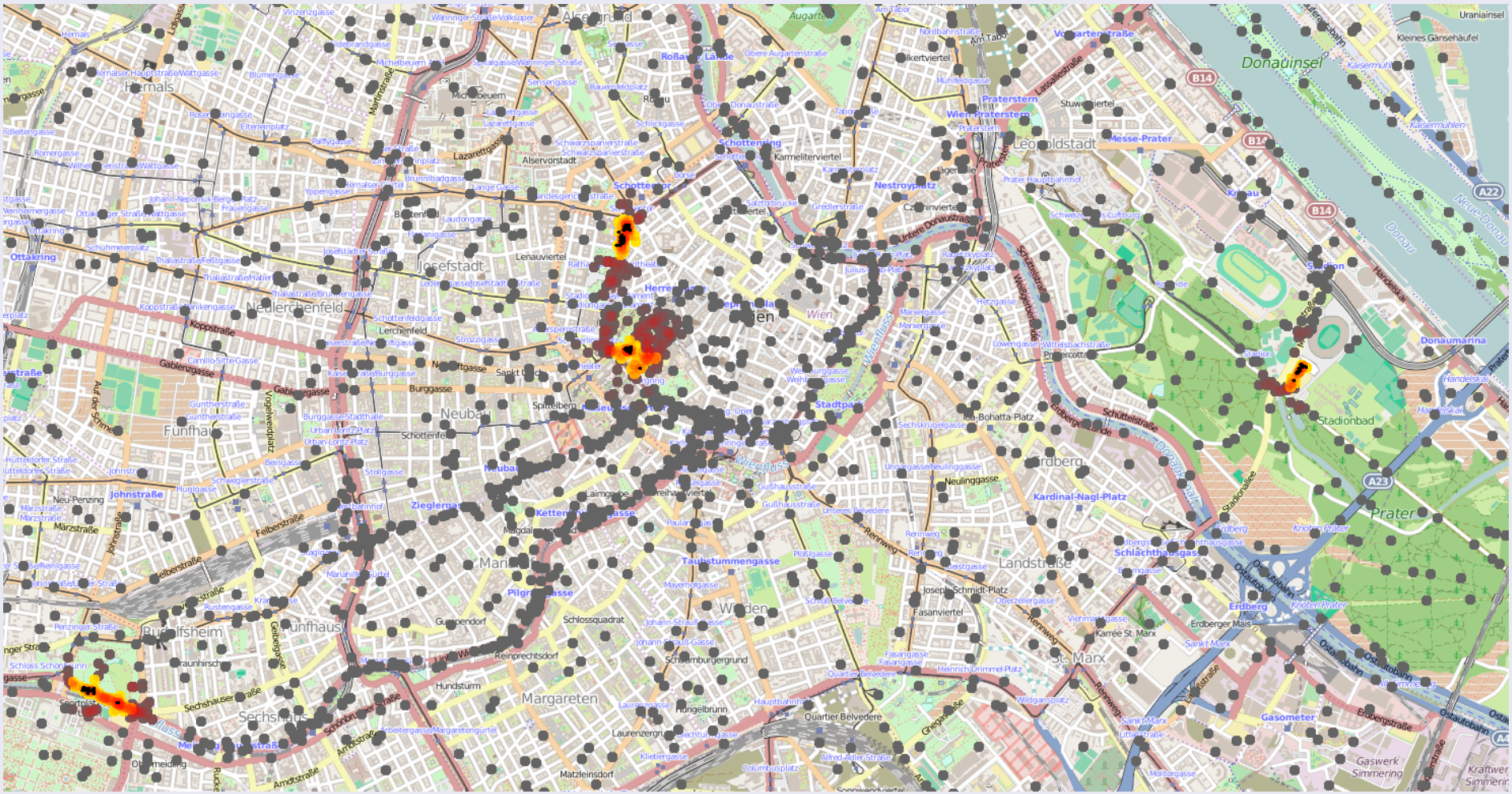
Crowd density estimation [BPV15]



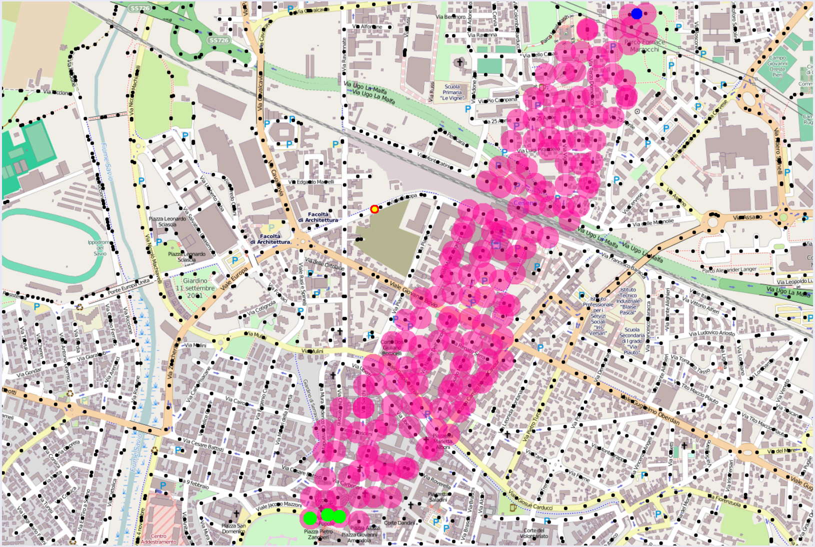
Metaprogramming: P2P version upgrade [BPV15]



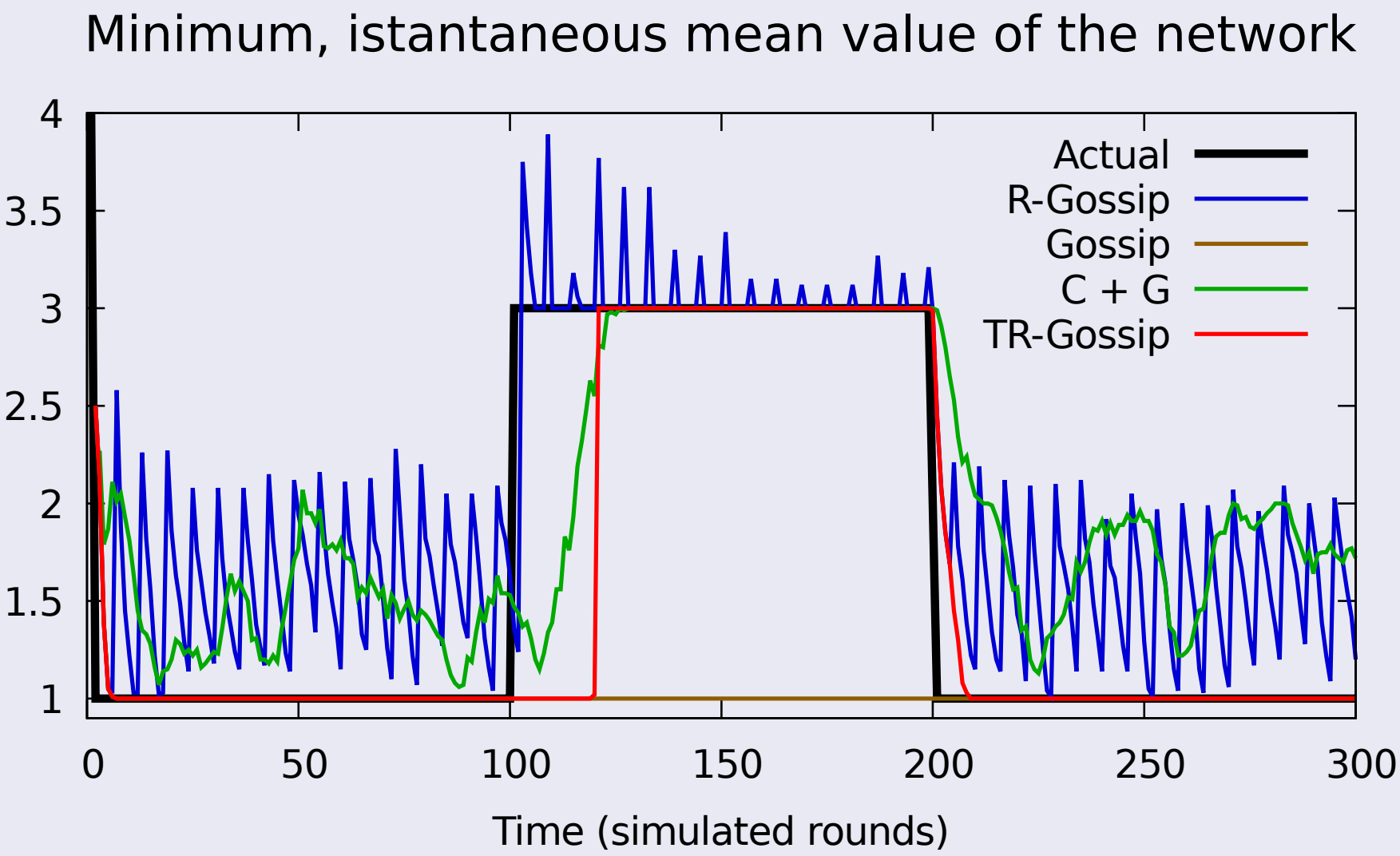
Metaprogramming: process priority modulation [BPV15]



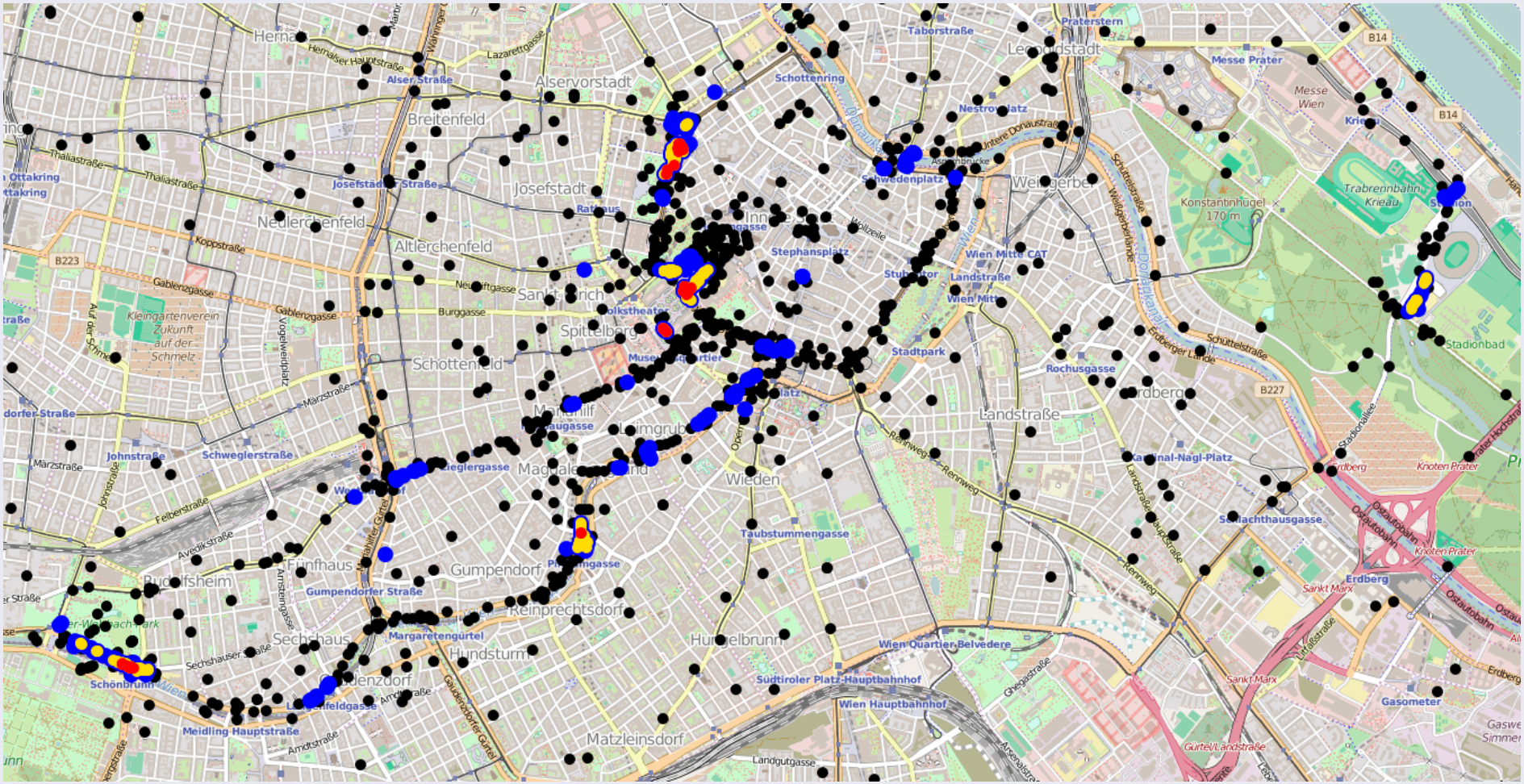
Resilient P2P communication with replicated channel [VPR+15, VPRC17]



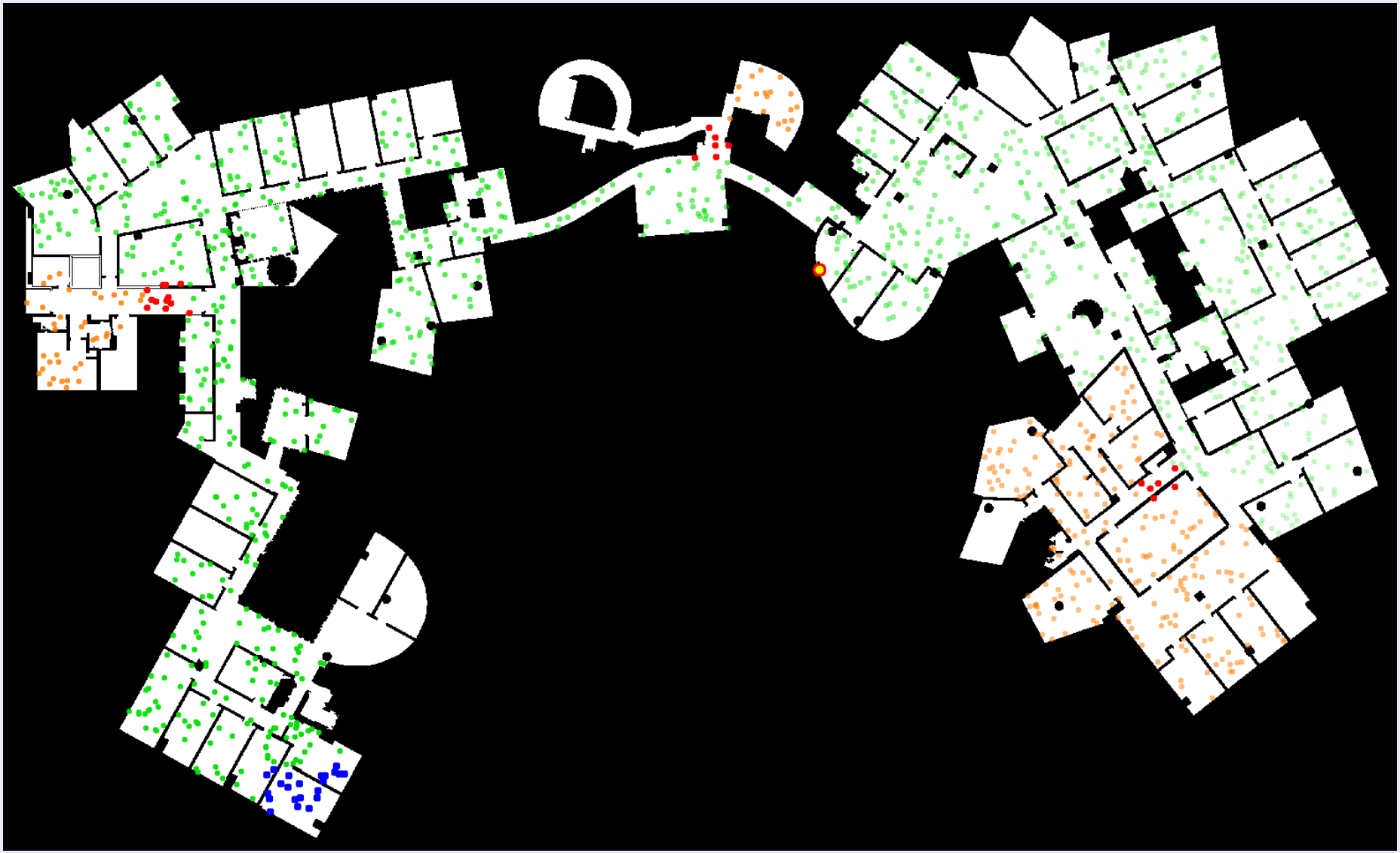
Metaprogramming: improved gossip dynamics [PBV16]



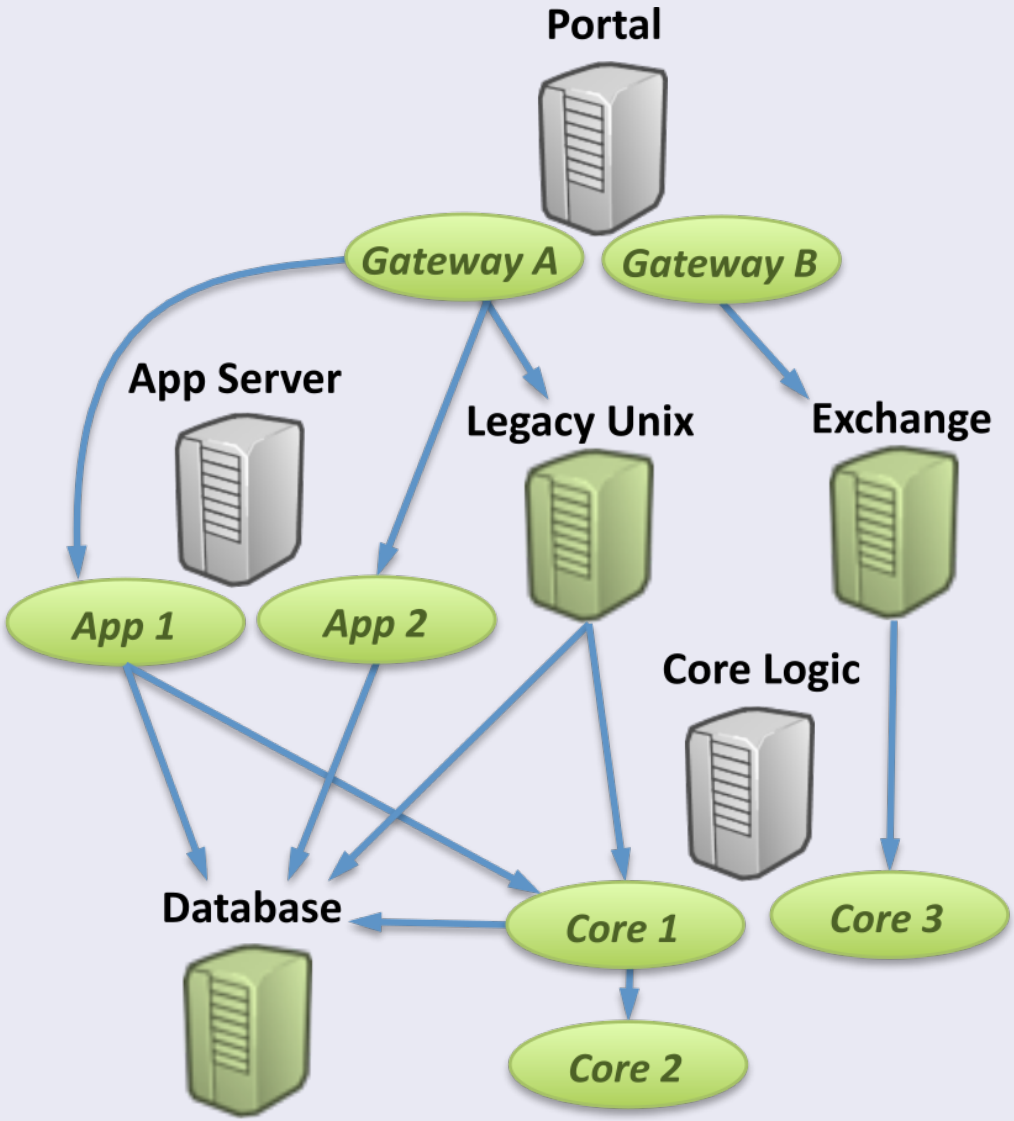
Crowd dispersal [VBPB16]



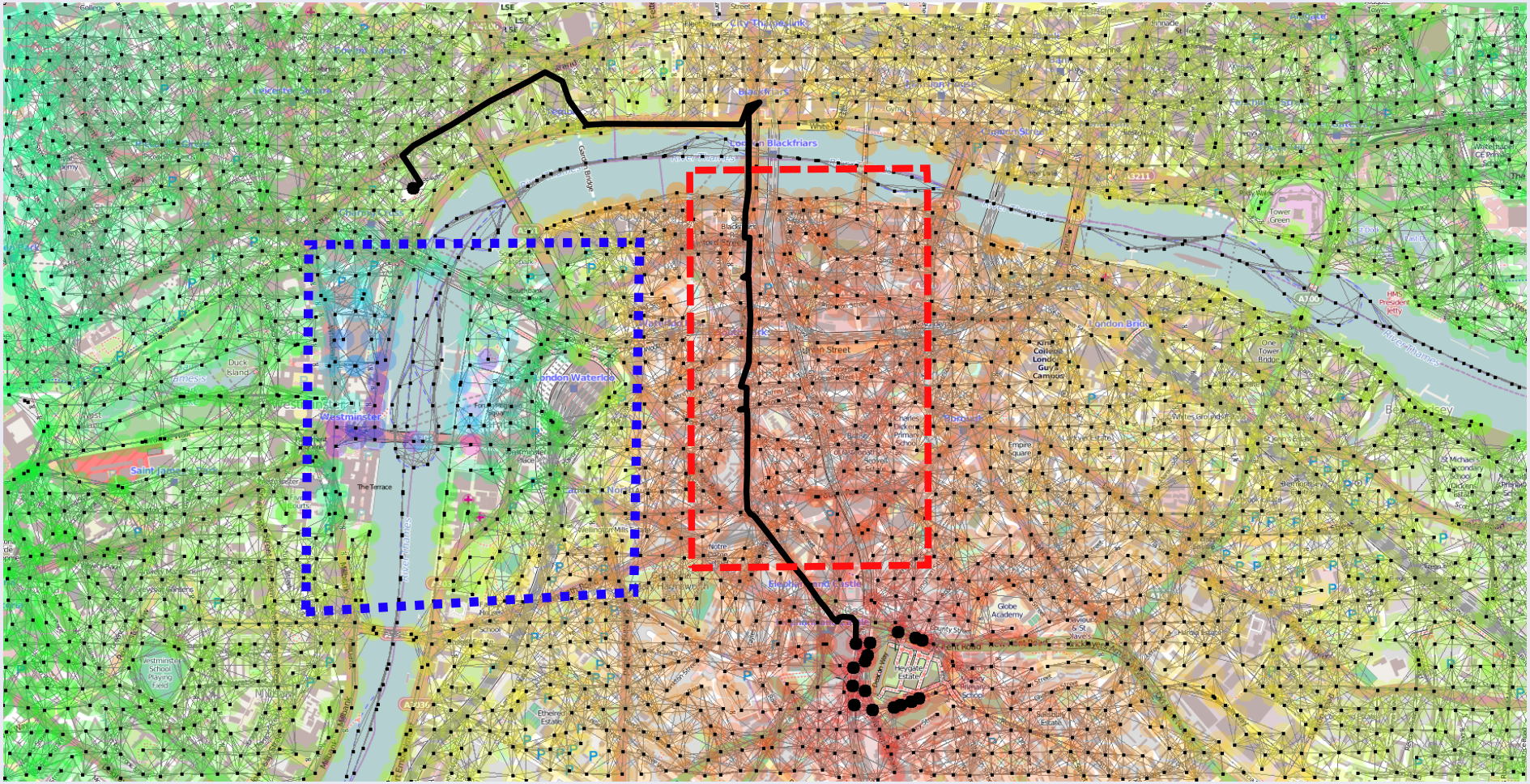
Indoor alarm and selective dispersal [VCP16b]



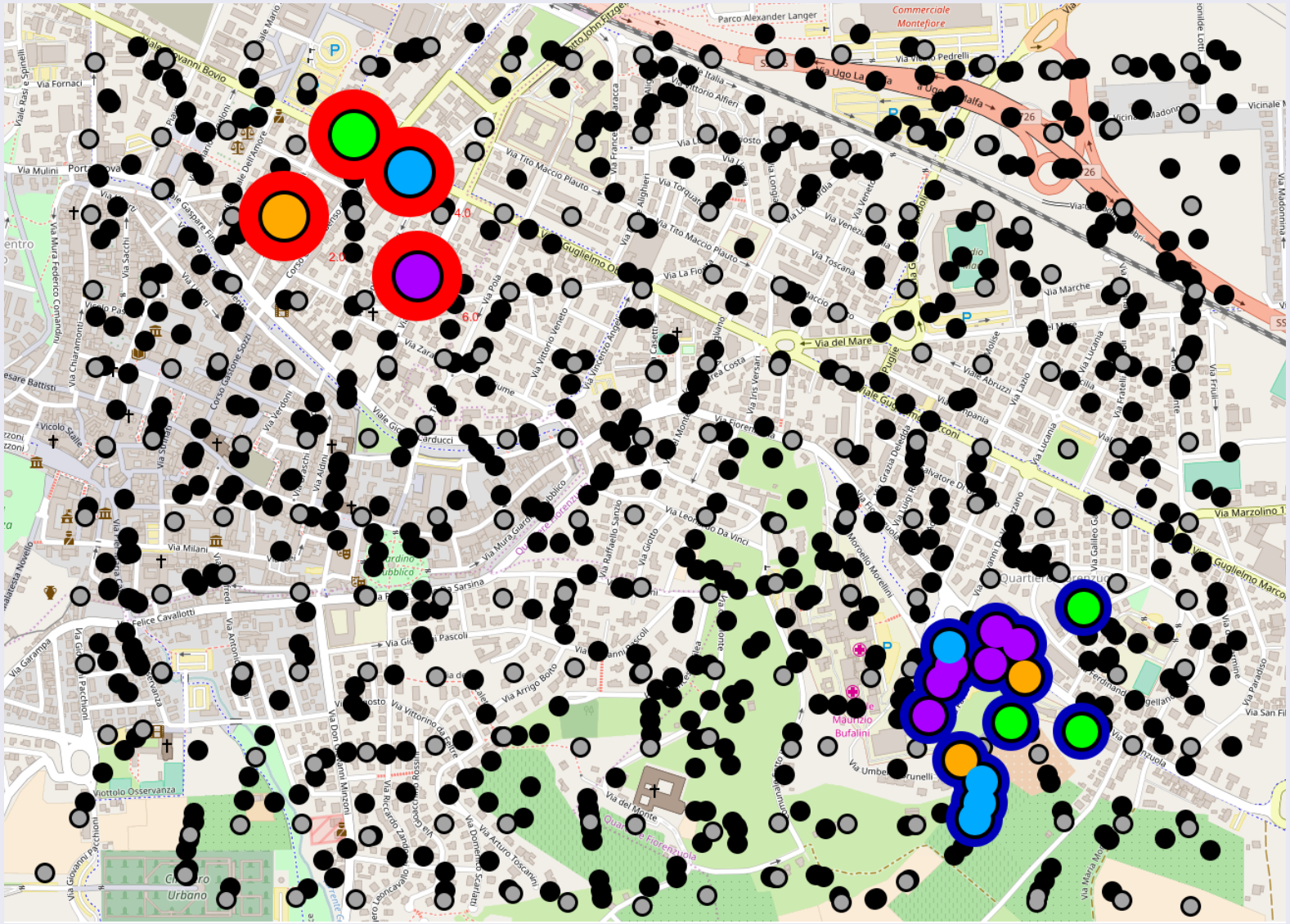
Distributed Recovery for Enterprise Services [CBP15]



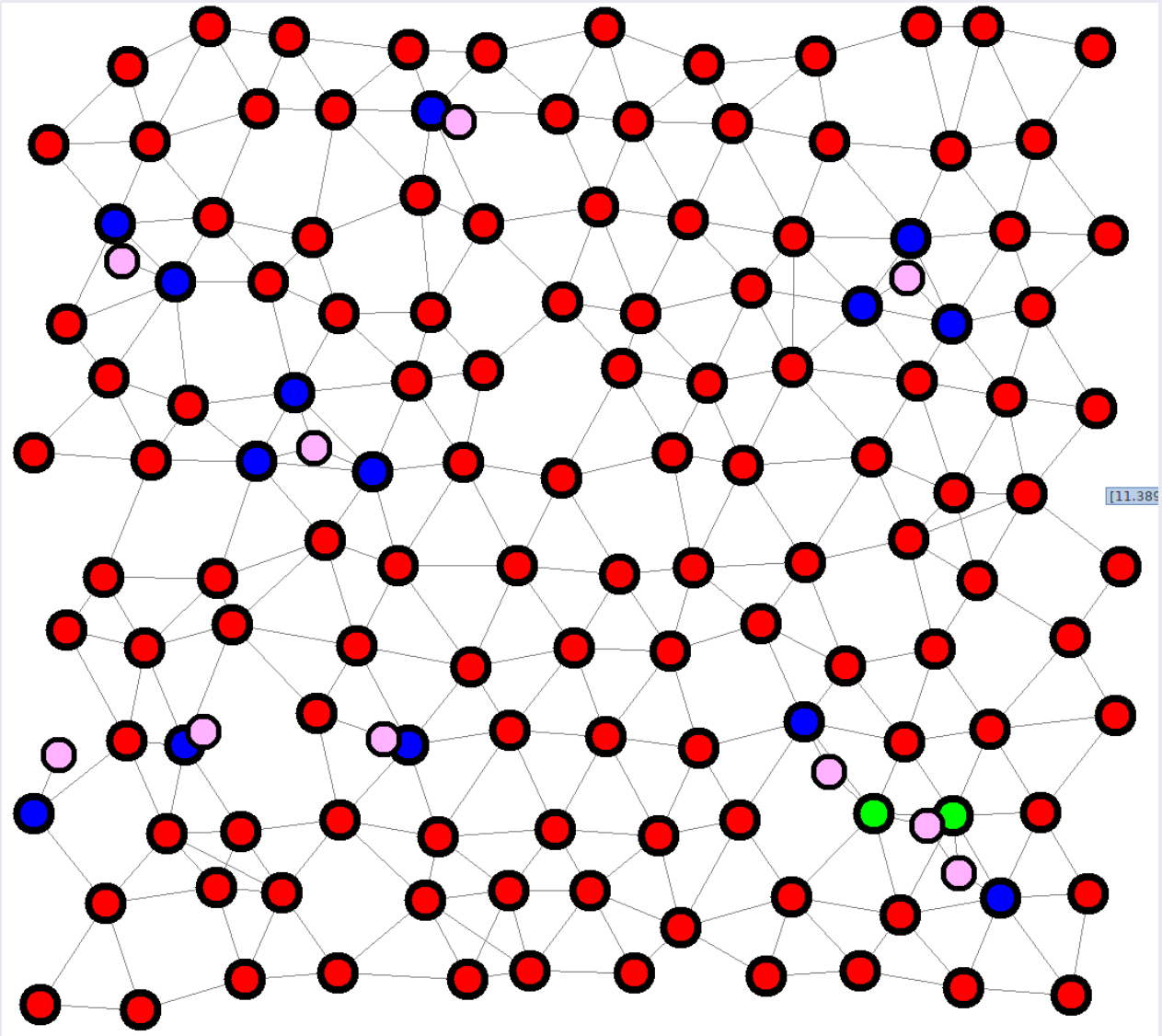
Context-sensitive steering [BVPD17, BVPD16]



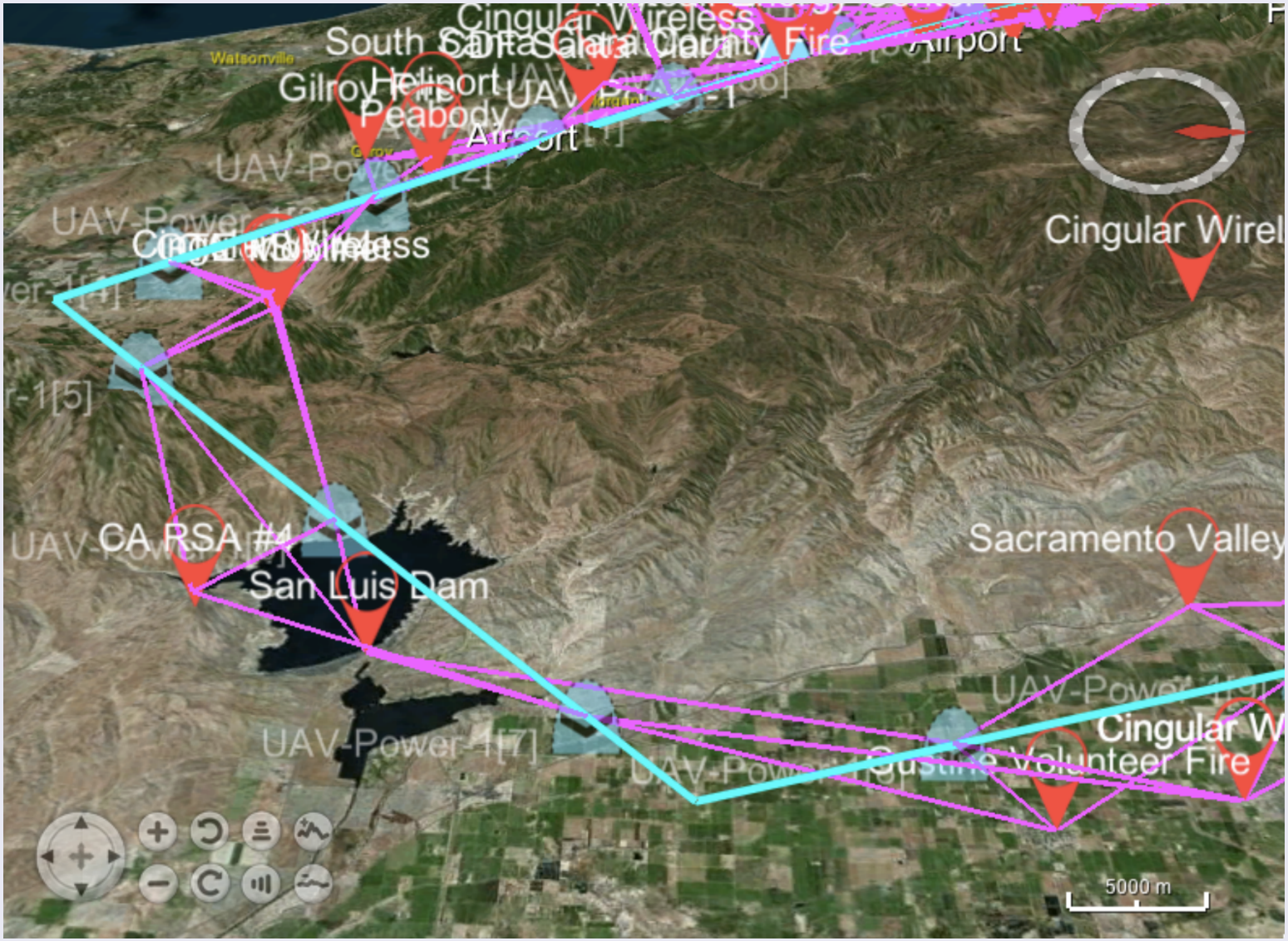
Smart allocation of limited resources [FPBV17]



Target counting in WSN [PDV17]

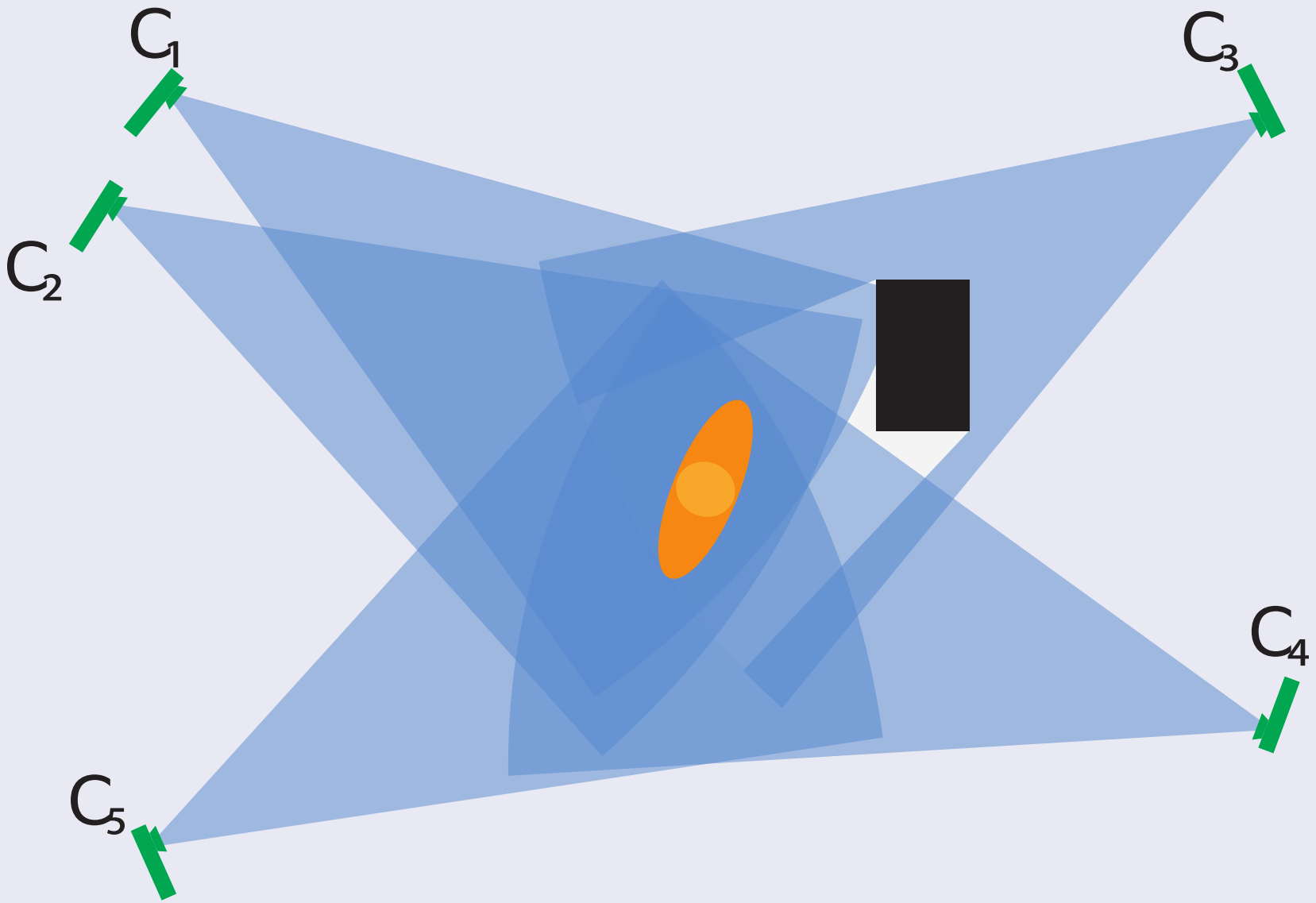


Airborne sensor sharing [BUL+18] and task reallocation [BUL+16]

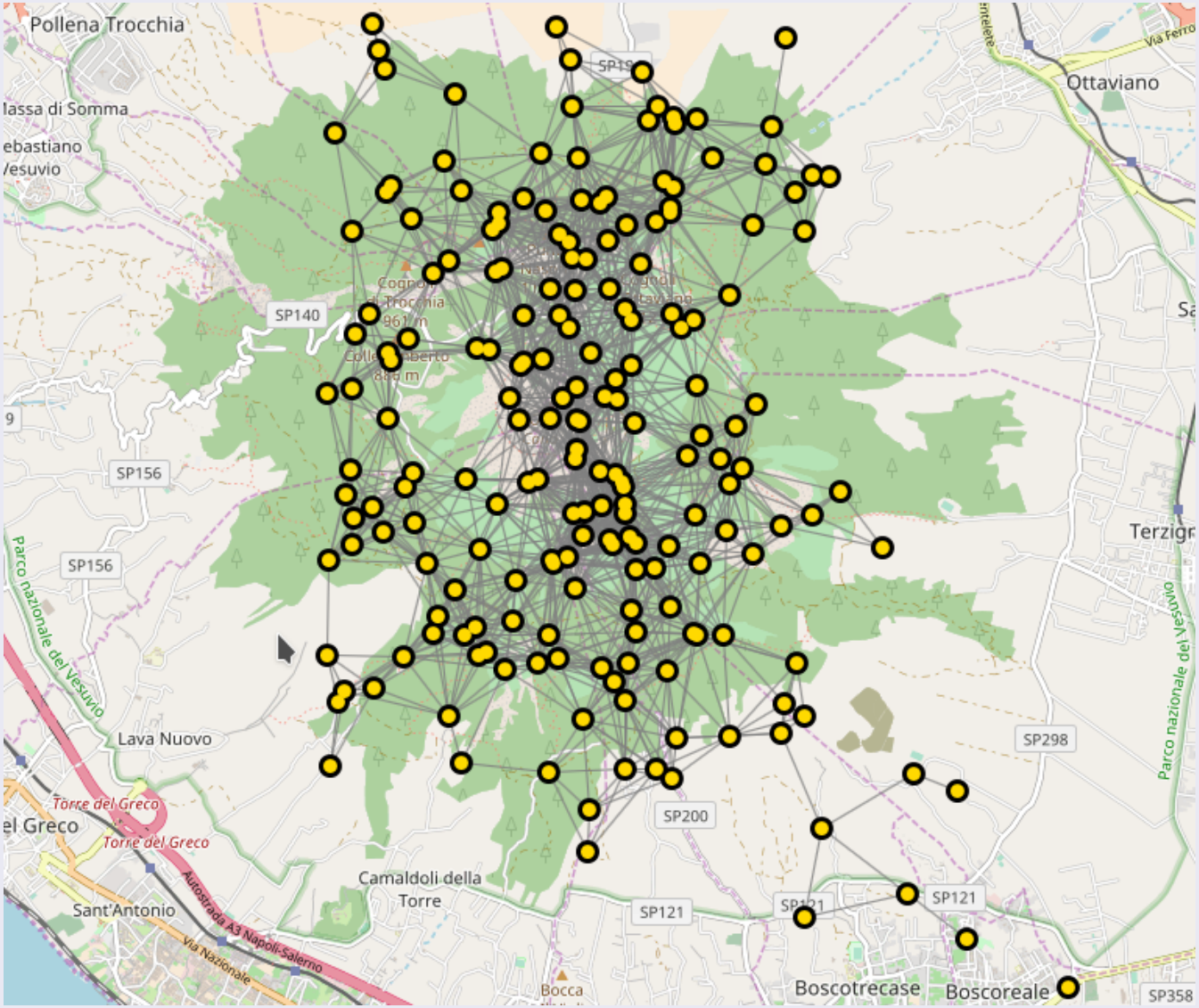


Outline

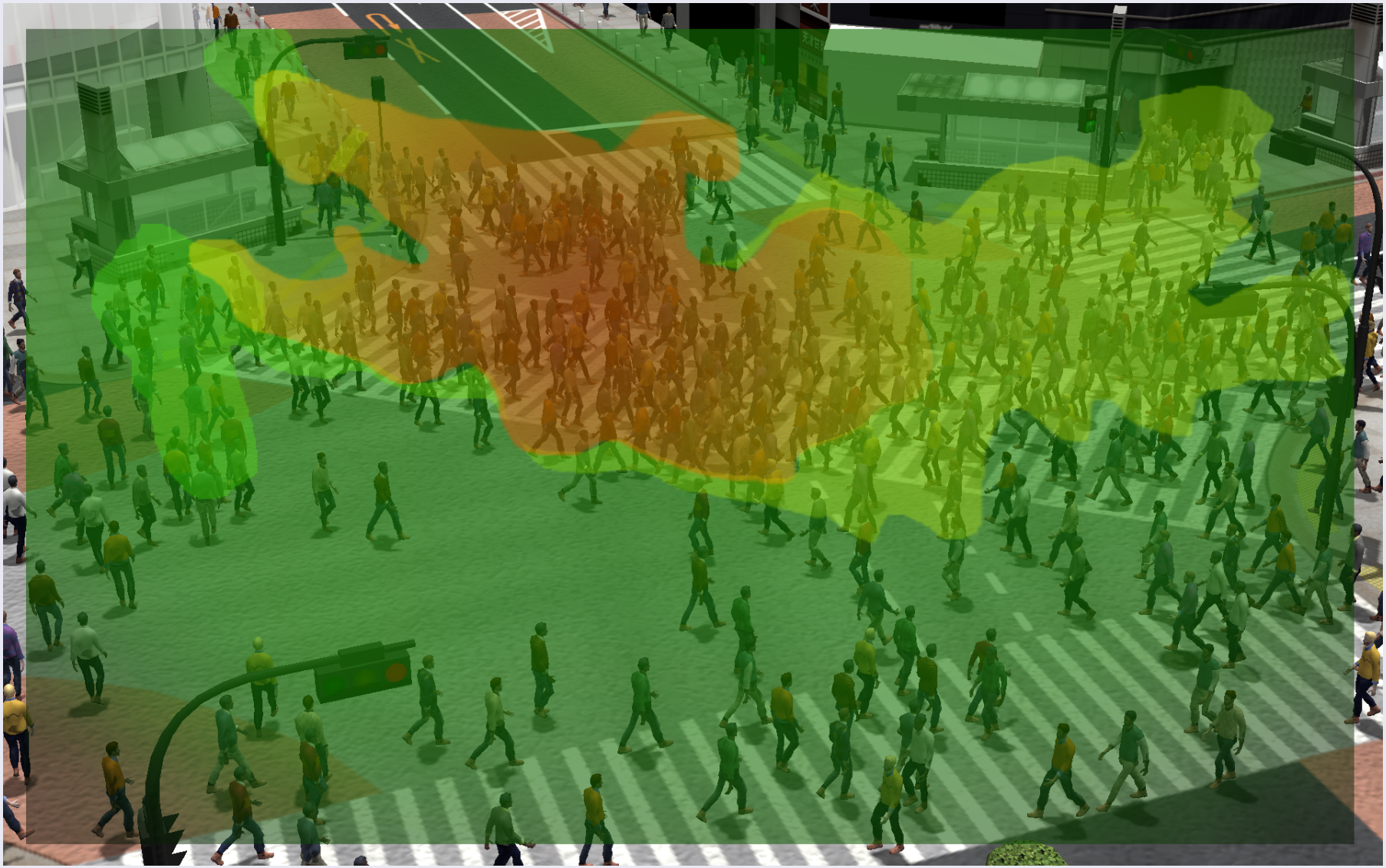
Smart cameras coordination [EL17]



Drone exploration



Augmented fields



Outline

Open issues I

Network overlays

- Aggregate programming relies on the concept of "neighbors" in order to build fields to compute upon
- The network of devices may (in opportunistic and P2P networks) or may not (in the classic Internet) reflect the actual network structure
- A better understanding of how to build network overlays for specific applications is required

Anonymous networks

- Aggregate programming is tightly bound to device identity
- An extension / restriction of the calculus for anonymous networks may expand the applicability of the approach

Open issues II

Security and privacy

- The kind of open systems aggregate computing promotes is a fertile ground for malicious attackers
  - Metaprogramming can be used to inject and diffuse malicious aggregate processes in the system rather efficiently
  - Byzantine attacks may disrupt entire algorithms
  - Malfunctioning nodes detection is (undesirably) offloaded to the application level
- Little work has been done on security [CAV17]

Open issues III

Cloud offloading

- Existing back-ends for aggregate programming do not take full advantage of cloud / fog resources
- Offloading part of the computation to the cloud when available may speed up algorithms and reduce bandwidth and battery usage
- Little has been done on this side [VCP16a]

Better understanding of aggregate alorithms through control theory

- Currently most algorithms are validated via simulation
- Possibly, a large number of them could be studied with a different perspective using control theory
- Much more solid results, better insights
- Very little work availabile [KBDM15]

Open issues IV

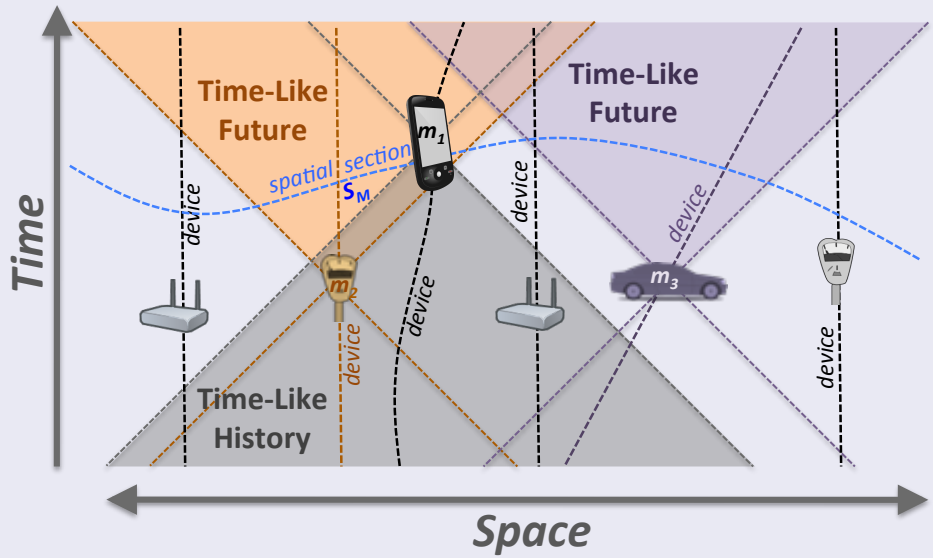
Limited resource scenarios

- None of the platforms for aggregate programming was conceived with a commitment to low resource usage
- Some early work exists to allow using LoRaWAN [All15] as network backend
- But packets sent by current implementations are huge and prevent practical use
- Strategies for reducing resource usage must be devised

Open issues V

"Computational relativity"

- Communication speed bounds the speed at which the various part of the aggregate machine perceives updates
- May be impacting on several deployments
- Somehow resembles the shift from classical mechanic to relativity



References I

LoRa Alliance.
A technical overview of lora and lorawan.
White Paper, November, 2015.

Jacob Beal, Danilo Pianini, and Mirko Viroli.
Aggregate programming for the internet of things.
IEEE Computer, 48(9):22–30, 2015.

Jacob Beal, Kyle Usbeck, Joseph P. Loyall, Mason Rowe, and James M. Metzler.
Adaptive task reallocation for airborne sensor sharing.
In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W), Augsburg, Germany, September 12-16, 2016*, pages 168–173, 2016.

Jacob Beal, Kyle Usbeck, Joseph P. Loyall, Mason Rowe, and James M. Metzler.
Adaptive opportunistic airborne sensor sharing.
TAAS, 13(1):6:1–6:29, 2018.

Jacob Beal and Mirko Viroli.
Building blocks for aggregate programming of self-organising applications.
In *Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014, London, United Kingdom, September 8-12, 2014*, pages 8–13, 2014.

References II

Jacob Beal, Mirko Viroli, Danilo Pianini, and Ferruccio Damiani.
Self-adaptation to device distribution changes.
In *10th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2016, Augsburg, Germany, September 12-16, 2016*, pages 60–69, 2016.

Jacob Beal, Mirko Viroli, Danilo Pianini, and Ferruccio Damiani.
Self-adaptation to device distribution in the internet of things.
ACM Trans. Auton. Adapt. Syst., 12(3):12:1–12:29, September 2017.

Roberto Casadei, Alessandro Aldini, and Mirko Viroli.
Combining trust and aggregate computing.
In *Software Engineering and Formal Methods - SEFM 2017 Collocated Workshops: DataMod, FAACS, MSE, CoSim-CPS, and FOCLASA, Trento, Italy, September 4-5, 2017, Revised Selected Papers*, pages 507–522, 2017.

Shane S. Clark, Jacob Beal, and Partha P. Pal.
Distributed recovery for enterprise services.
In *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems, Cambridge, MA, USA, September 21-25, 2015*, pages 111–120, 2015.

References III

Roberto Casadei, Giancarlo Fortino, Danilo Pianini, Wilma Russo, Claudio Savaglio, and Mirko Viroli.
Modelling and simulation of opportunistic iot services with aggregate computing.
Future Generation Computer Systems, 2018.
Submitted for publication.

Roberto Casadei and Mirko Viroli.
Towards aggregate programming in scala.
In *First Workshop on Programming Models and Languages for Distributed Computing, PMLDC@ECOOP 2016, Rome, Italy, July 17, 2016*, page 5, 2016.

Simon Dobson, Mirko Viroli, Jose Luis Fernandez-Marquez, Franco Zambonelli, Graeme Stevenson, Giovanna Di Marzo Serugendo, Sara Montagna, Danilo Pianini, Juan Ye, Gabriella Castelli, and et al.
Spatial awareness in pervasive ecosystems.
The Knowledge Engineering Review, 31(4):343–366, Sep 2016.

References IV


Ferruccio Damiani, Mirko Viroli, Danilo Pianini, and Jacob Beal.
Code mobility meets self-organisation: A higher-order calculus of computational fields.
In *Formal Techniques for Distributed Objects, Components, and Systems - 35th IFIP WG 6.1 International Conference, FORTE 2015, Held as Part of the 10th International Federated Conference on Distributed Computing Techniques, DisCoTec 2015, Grenoble, France, June 2-4, 2015, Proceedings*, pages 113–128, 2015.


Lukas Esterle and Peter R. Lewis.
Online multi-object k-coverage with mobile smart cameras.
In *Proceedings of the 11th International Conference on Distributed Smart Cameras, Stanford, CA, USA, September 5-7, 2017*, pages 107–112, 2017.


Matteo Francia, Danilo Pianini, Jacob Beal, and Mirko Viroli.
Towards a foundational API for resilient distributed systems design.
In *2nd IEEE International Workshops on Foundations and Applications of Self* Systems, FAS*W@SASO/ICCAC 2017, Tucson, AZ, USA, September 18-22, 2017*, pages 27–32, 2017.

References V


Amy Kumar, Jacob Beal, Soura Dasgupta, and Raghu Mudumbai.
Toward predicting distributed systems dynamics.
In *2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASO Workshops 2015, Cambridge, MA, USA, September 21-25, 2015*, pages 68–73, 2015.


Danilo Pianini, Jacob Beal, and Mirko Viroli.
Improving gossip dynamics through overlapping replicates.
In *Coordination Models and Languages - 18th IFIP WG 6.1 International Conference, COORDINATION 2016, Held as Part of the 11th International Federated Conference on Distributed Computing Techniques, DisCoTec 2016, Heraklion, Crete, Greece, June 6-9, 2016, Proceedings*, pages 192–207, 2016.


Danilo Pianini, Jacob Beal, and Mirko Viroli.
Practical aggregate programming with protelis.
In *2nd IEEE International Workshops on Foundations and Applications of Self* Systems, FAS*W@SASO/ICCAC 2017, Tucson, AZ, USA, September 18-22, 2017*, pages 391–392, 2017.

References VI


Danilo Pianini, Angelo Croatti, Alessandro Ricci, and Mirko Viroli.
Computational fields meet augmented reality: Perspectives and challenges.
In *2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASO Workshops 2015, Cambridge, MA, USA, September 21-25, 2015*, pages 80–85, 2015.


Danilo Pianini, Simon Dobson, and Mirko Viroli.
Self-stabilising target counting in wireless sensor networks using euler integration.
In *11th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2017, Tucson, AZ, USA, September 18-22, 2017*, pages 11–20, 2017.


Danilo Pianini, Sara Montagna, and Mirko Viroli.
Chemical-oriented simulation of computational systems with ALCHEMIST.
J. Simulation, 7(3):202–215, 2013.


Danilo Pianini, Mirko Viroli, and Jacob Beal.
Protelis: practical aggregate programming.
In *Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, April 13-17, 2015*, pages 1846–1853, 2015.

References VII

Alessandro Ricci, Mirko Viroli, Andrea Omicini, Stefano Mariani, Angelo Croatti, and Danilo Pianini.
Spatial tuples: Augmenting reality with tuples.
Expert Systems, page e12273, apr 2018.

Mirko Viroli, Giorgio Audrito, Jacob Beal, Ferruccio Damiani, and Danilo Pianini.
Engineering resilient collective adaptive systems by self-stabilisation.
ACM Transactions on Modeling and Computer Simulation, 28(2):1–28, mar 2018.

Mirko Viroli, Giorgio Audrito, Ferruccio Damiani, Danilo Pianini, and Jacob Beal.
A higher-order calculus of computational fields.
CoRR, abs/1610.08116, 2016.

Mirko Viroli, Giorgio Audrito, Ferruccio Damiani, Danilo Pianini, and Jacob Beal.
A higher-order calculus of computational fields.
ACM Transactions on Computational Logic, 2017.
Submitted for publication.

References VIII

Mirko Viroli, Jacob Beal, Ferruccio Damiani, Giorgio Audrito, Roberto Casadei, and Danilo Pianini.
From field-based coordination to aggregate computing.
In *Coordination Models and Languages 2018. Proceedings*, 2018.
To appear.

Mirko Viroli, Jacob Beal, Ferruccio Damiani, and Danilo Pianini.
Efficient engineering of complex self-organising systems by self-stabilising fields.
In *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems, Cambridge, MA, USA, September 21-25, 2015*, pages 81–90, 2015.

Mirko Viroli, Antonio Bucchiarone, Danilo Pianini, and Jacob Beal.
Combining self-organisation and autonomic computing in cass with aggregate-mape.
pages 186–191, 2016.

Mirko Viroli, Roberto Casadei, and Danilo Pianini.
On execution platforms for large-scale aggregate computing.
In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp Adjunct 2016, Heidelberg, Germany, September 12-16, 2016*, pages 1321–1326, 2016.

References IX


Mirko Viroli, Roberto Casadei, and Danilo Pianini.
Simulating large-scale aggregate mass with alchemist and scala.
In *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016.*, pages 1495–1504, 2016.


Mirko Viroli, Danilo Pianini, and Jacob Beal.
Linda in space-time: An adaptive coordination model for mobile ad-hoc environments.
In *Coordination Models and Languages - 14th International Conference, COORDINATION 2012, Stockholm, Sweden, June 14-15, 2012. Proceedings*, pages 212–229, 2012.


Mirko Viroli, Danilo Pianini, Alessandro Ricci, Pietro Brunetti, and Angelo Croatti.
Multi-agent systems meet aggregate programming: Towards a notion of aggregate plan.
In *PRIMA 2015: Principles and Practice of Multi-Agent Systems - 18th International Conference, Bertinoro, Italy, October 26-30, 2015, Proceedings*, pages 49–64, 2015.


Mirko Viroli, Danilo Pianini, Alessandro Ricci, and Angelo Croatti.
Aggregate plans for multiagent systems.
International Journal of Agent-Oriented Software Engineering, 5(4):336, 2017.