Towards Computing Without Borders: A Data Processing Plane

Reginald Cushing[1], Marian Bubak[1,2], Adam Belloum[1], Marc X. Makkes[1], Cees de Laat[1]

[1]Informatics Institute, University of Amsterdam, Science Park 904, 1098 XH, Amsterdam, Netherlands
[2]AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Krakow

Motivation

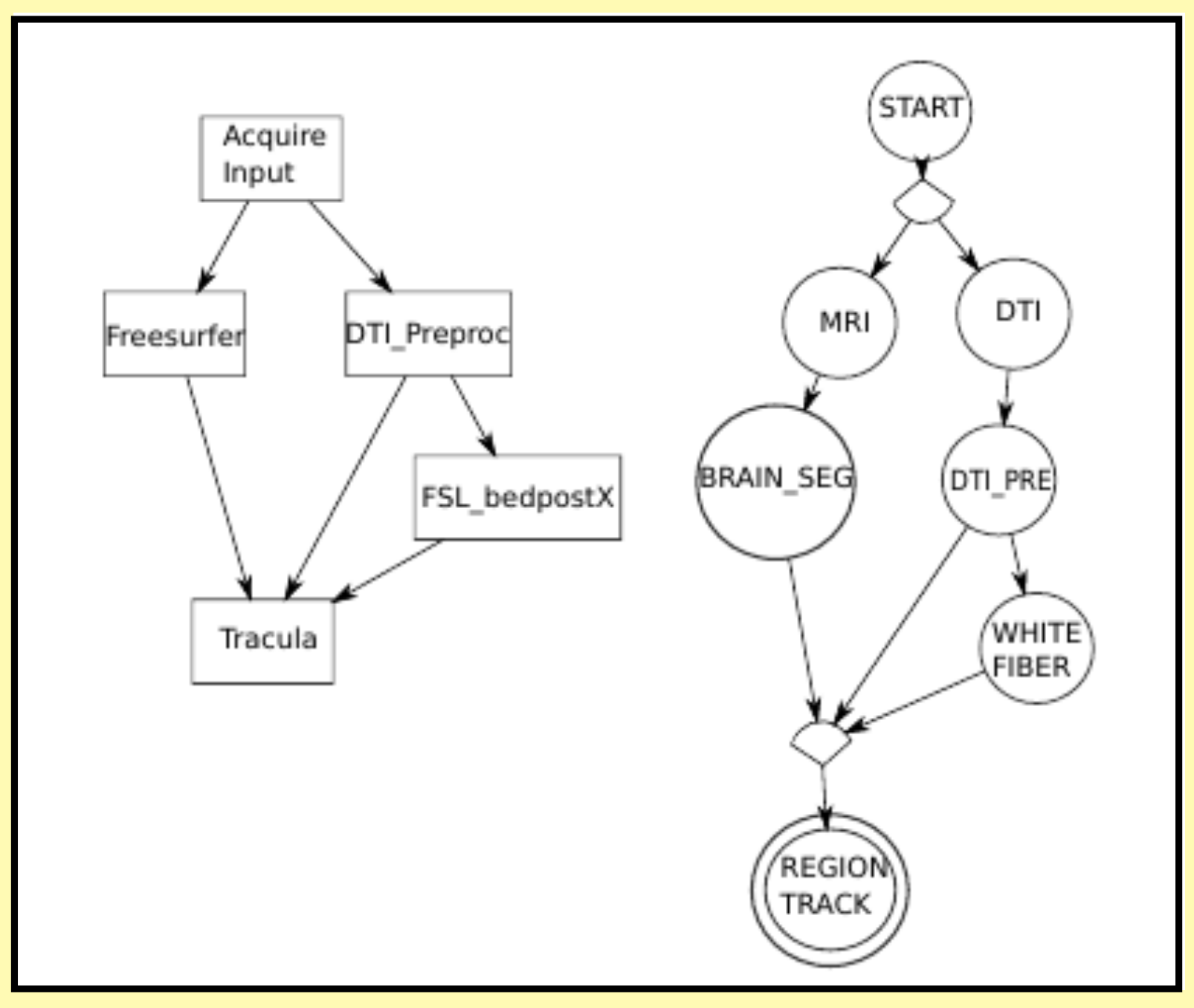
Data partitionability, processing complexity and locality play a crucial role in the effectiveness of distributed systems. Through virtualization, resources have become scattered, heterogeneous, and dynamic in performance and networking. Collective and collaborative use of these resources for data processing is our main challenge.

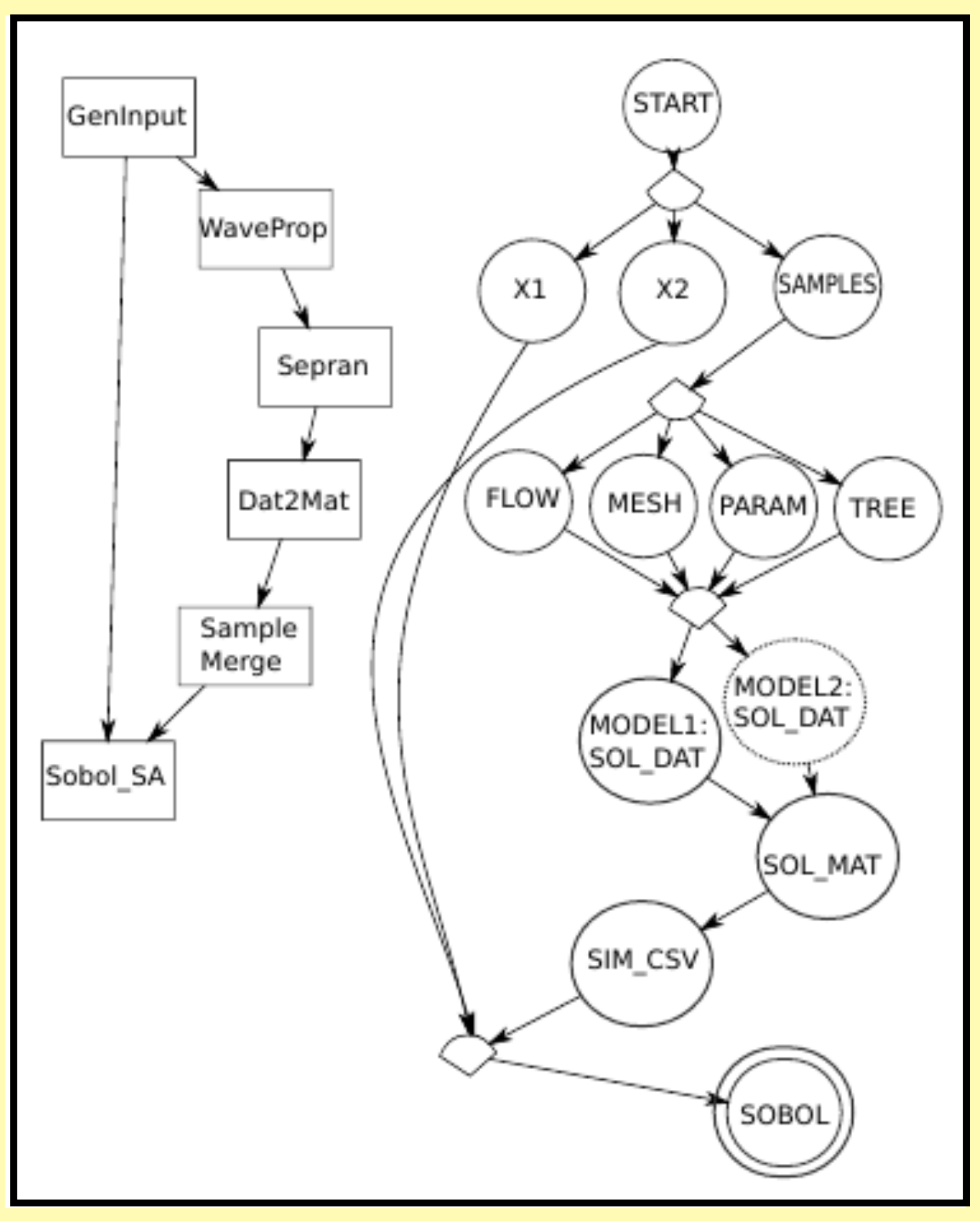
Automata as a Data Processing Schema

Automata is an intuitive way to describe data processing a a transformation from one state to the next. The data transformation model can be considered as a 5 tuple NFA:

$$(Q, \Sigma, \delta, q_0, F)$$

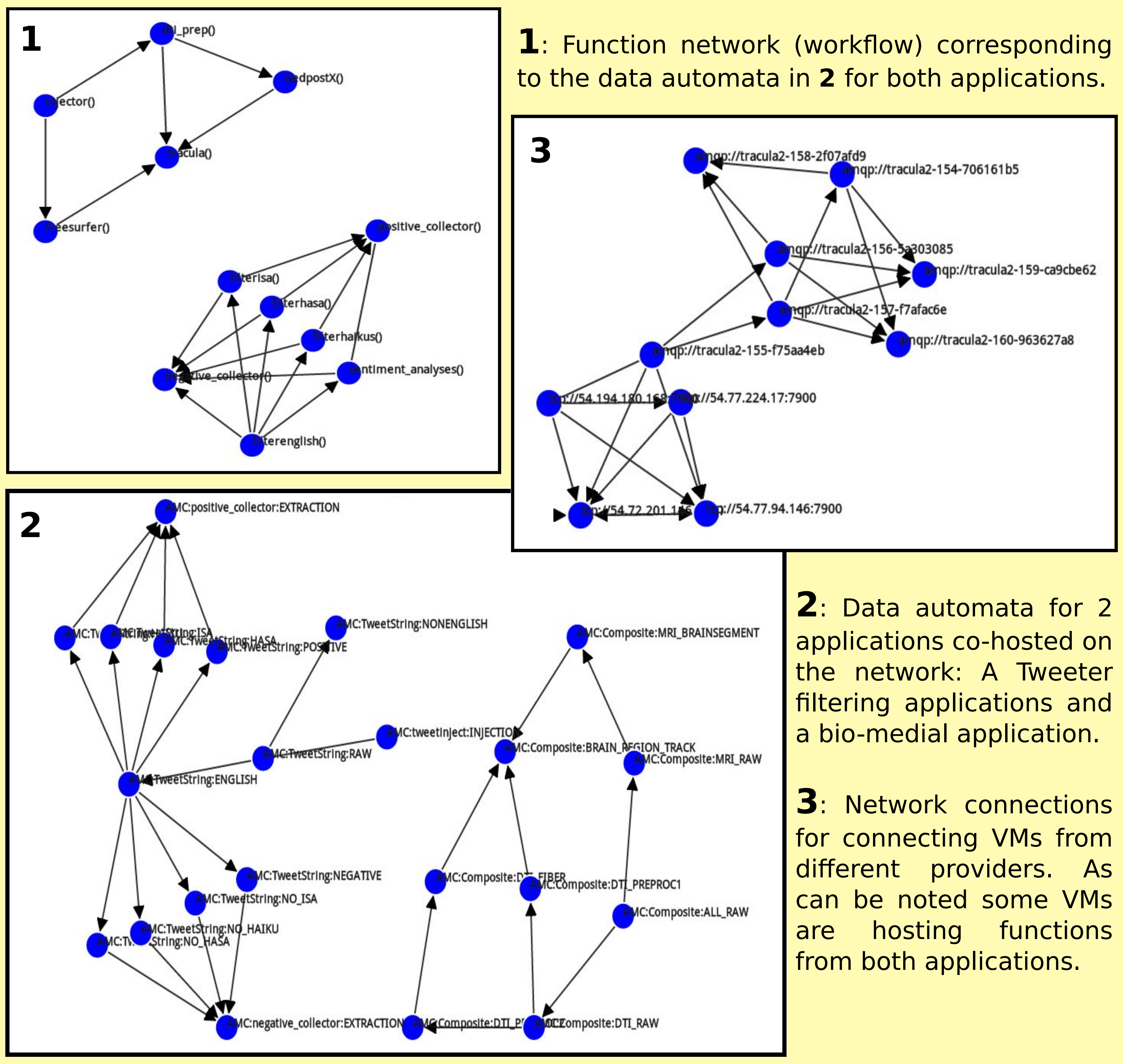
Q is the set of states the data object can be in. **Σ** is the set of functions that performs the data transformations. **δ** is the transition function that maps data and functions to new states such that **Q x Σ -> P(Q)**. **F** is the set of final data states which mark the completion of processing. **q_0** is a starting data state.





Left: Bio-medical workflow and its corresponding data automaton. **Right**: Blood flow simulation workflow and data automaton.

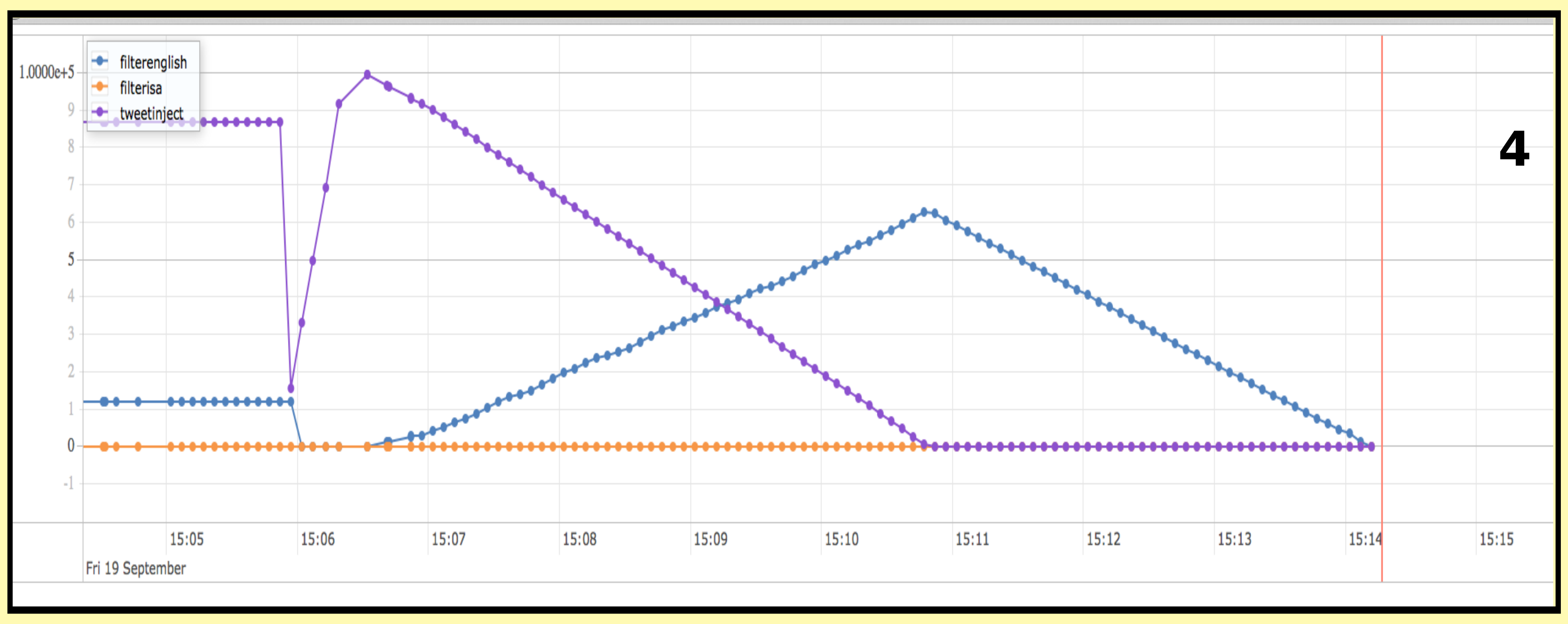
PUMPKIN in Action



1: Function network (workflow) corresponding to the data automata in **2** for both applications.

2: Data automata for 2 applications co-hosted on the network: A Tweeter filtering applications and a bio-medial application.

3: Network connections for connecting VMs from different providers. As can be noted some VMs are hosting functions from both applications.



4: Live processing and networking statistics from individual processing functions.

Background

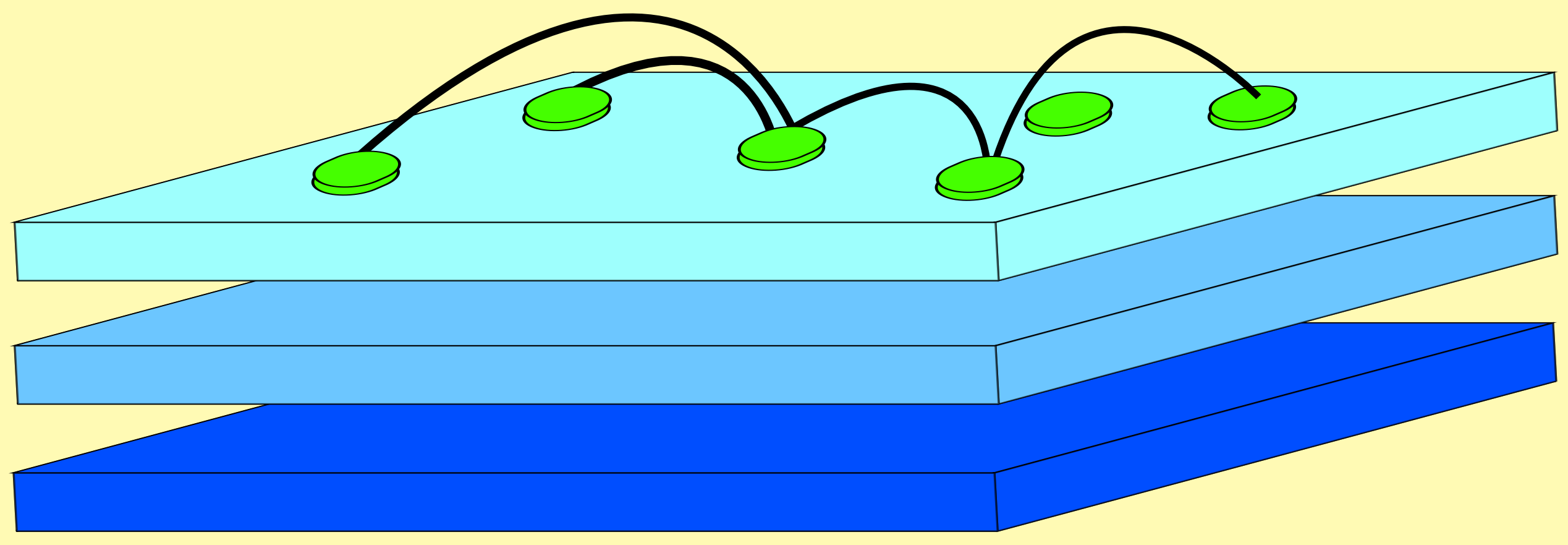
In eScience, coordinating multiple tasks for running *in-silico* experiments is often the realm of Scientific Workflow Management Systems (SWMS). These are often centralized systems that work in confined resources.

A common denominator in most workflow systems is that the unit of reason is the process i.e. the abstract workflow describes a topology of tasks configured in a certain way. This is often tailored to the underlying infrastructure. **Thus the process ordering is a description of how to best exploit resources and not necessarily a description of data processing.**

The complexity and dynamism in big data processing entails a **new unit of reason: the data itself**. An abstract model for data processing will solely describe data transformations agnostically from the underlying resources.

Distributed Data Processing as a Protocol

Automata data model describes the abstract data processing model. The same model is used to build a distributed processing infrastructure around the data processing schema. The schema represents the knowledge of how data can be processed which at a network and resource level this represents a **data routing table**.



Data Networks:- Data Processing

Soft Networks:- SDNs

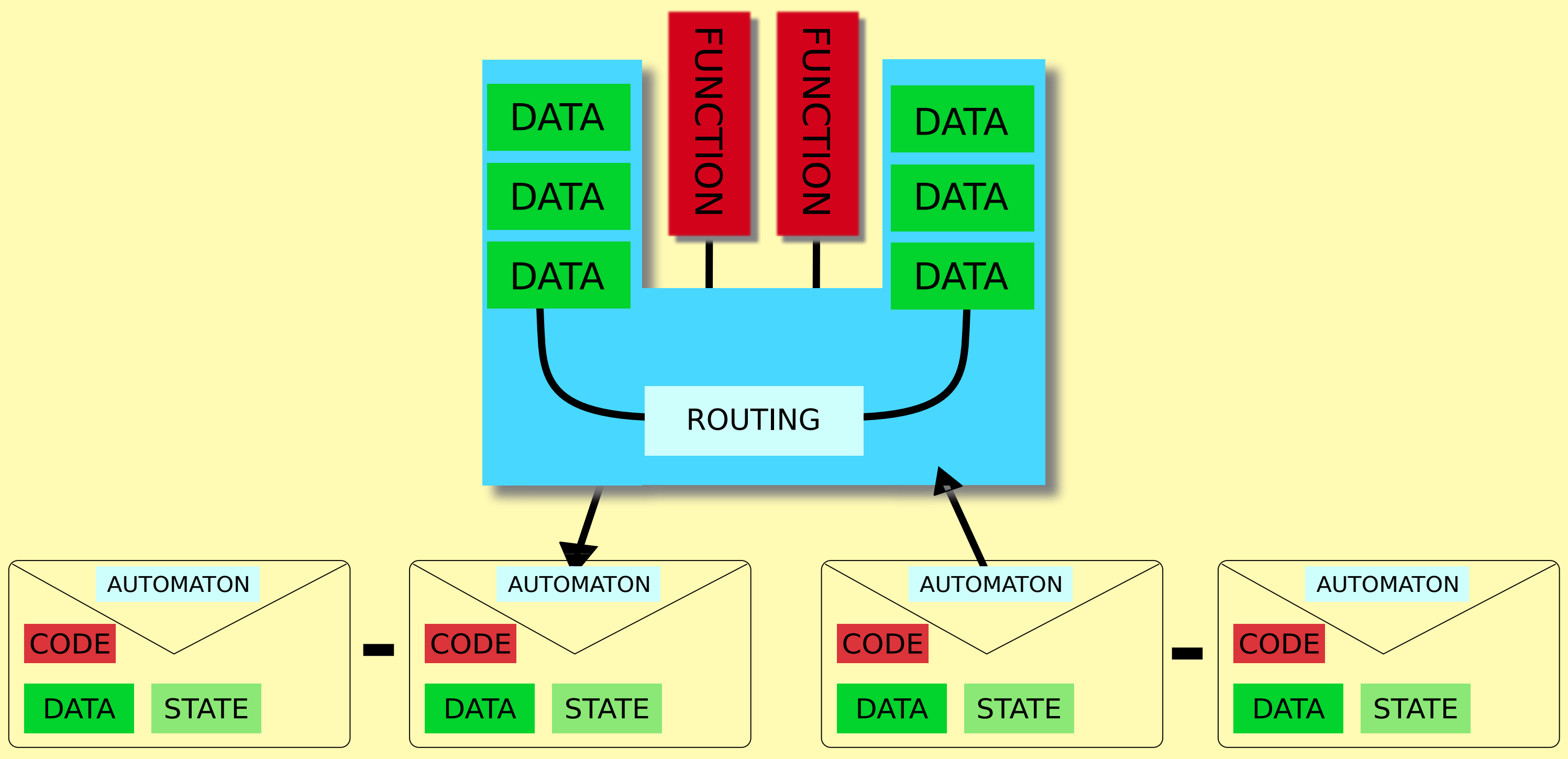
Hard Networks:- physical networks

Globally distributed resources are combined together in the **PUMPKIN** framework through a data processing protocol. Data is partitioned into packets. A packet is an **atomic unit of data processing**. Each data packet can encapsulate the automata as part of the header. The automaton header makes the packet self aware of where it has to go for processing. The data packet can also contain the code for processing the packet.

Data Packet = Data + Automaton + Code + State

The processing granularity is at the data packet level. This allows for various controllability such as **scaling** at packet level. A packet source will load balance data packets to multiple replicated, identical data processing functions. **Replication** is also at the packet level. Data packets can be replicated to multiple functions requesting the same data state.

Data processing functions are hosted in nodes. Functions can be statically deployed or deployed through the data packet since the packet can also carry code. The task for each node is two fold: nodes **process** data and also **route** data.



Each node in PUMPKIN **discovers** routes to other nodes. A routing table allows nodes to send data packets to the next node in a **P2P** fashion. In SDNs the routing table can be used to reconfigure the network.

References

[1] R. Cushing, A.S.Z. Belloum, M. Bubak, C. Laat, Automata-based Dynamic Data Processing for Clouds, In *Euro-Par 2014 Workshop Proceedings*, 2014.
[2] R. Cushing, S. Koulouzis, A.S.Z. Belloum, M. Bubak. Dynamic Handling for Cooperating Scientific Web Services. In *IEEE 7th International Conference on E-Science*, 2011.
[3] R. Cushing, S. Koulouzis, A.S.Z Belloum, and M. Bubak. Applying workflow as a service paradigm to application farming. *Concurrency and Computation: Practice and Experience*, 26(6):1297–1312, 2014.

Acknowledgments: This work is funded by the COMMIT (http://commit-nl.nl/) project.