



# Adaptation Mechanisms in P2P Networks

A multi-agent simulation using Repast

**Antonio Bucchiarone**

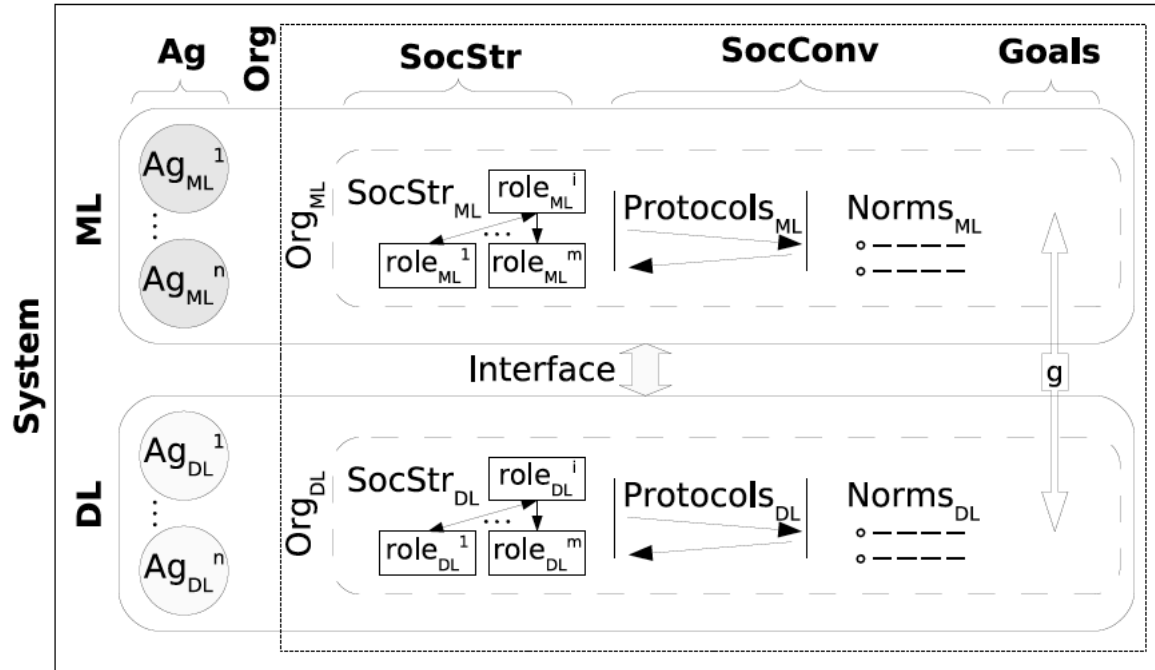
Fondazione Bruno Kessler, Trento – Italy

[bucchiarone@fbk.eu](mailto:bucchiarone@fbk.eu)

28 October 2019

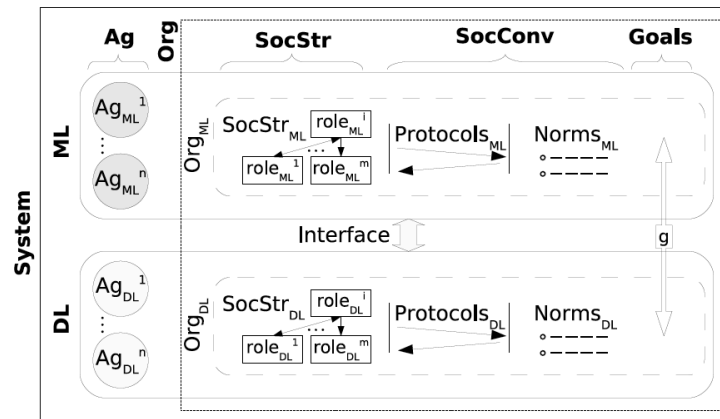
- **Adapting organisational structures** to maintain an *organisation effectiveness* under varying circumstances.
- **Population changes** may imply a decrease in **goal fulfilment**.
- **Open MAS**: we can not guarantee that agents are endowed with organisational adaptation capabilities.
- **Adaptation as an assistance to agents** that can be provided by MAS infrastructure.
- Two Level Assisted MAS Architecture. (2-LAMA).
- A simulator for testing **organisational adaptation mechanisms in P2P scenarios**.

- Domains where organisations can be dynamically changed
  - **Peer Sharing Networks (P2P)**
- Computers contact other computers to share some data
  - their relationships change over time depending on *network status* and *participants*.
- A **P2P system** is modelled as an organisation having:
  - a social structure among peers and,
  - a set of protocols and norms that regulate the sharing process.
- On top of the P2P system (Domain-Level) we add a distributed meta-level that:
  - *perceives status information* and uses it to **adapt peers social structure** and **norm values**.



- **Organisational Layer:** it provides coordination enabling services.
- **Assistance Layer:** it provides coordination assistance services.
  - Proactive service that adapts organisations depending on system's evolution.

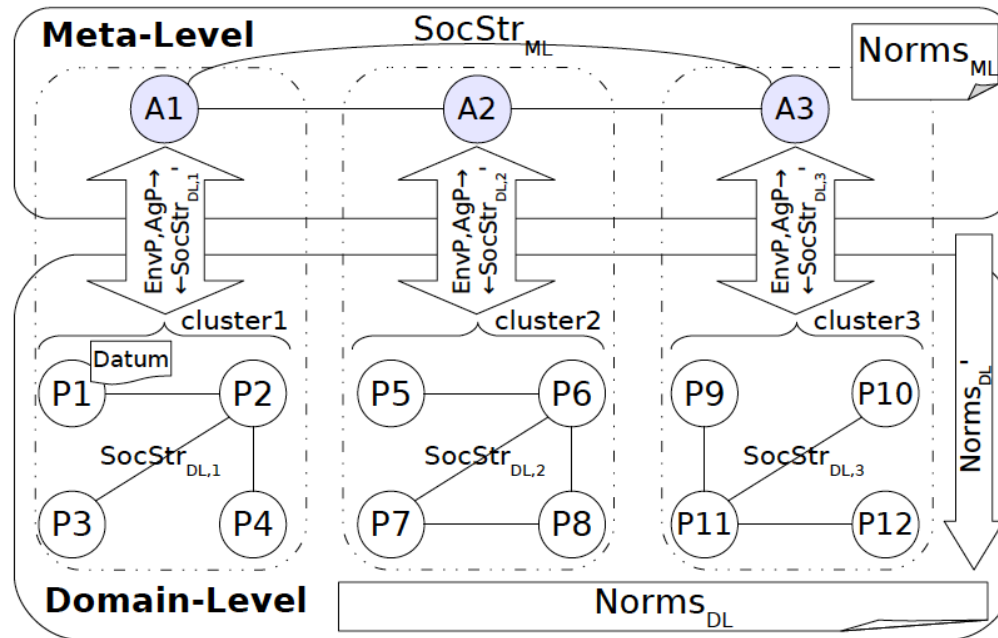
- $Org = \langle SocStr, SocConv, Goals \rangle$
- **SocStr**: social structure, which consists of a set of roles and relationships among them.
- **SocConv**: social conventions that agents should confirm and expect others to conform.
  - **Norms and/or interaction protocols**: *sequence of actions* performed by agents playing certain roles.
- **Goals**: Organisational design purpose.
  - Function or observable properties so that the system can evaluate its own performance.



- It provides two main types of services:
  - **Agent Assistance:** it assists individual agents to follow current social conventions.
    - *Information* for participating in the MAS
    - *Justification* of specific actions' consequences.
    - *Advice* of alternative plans conforming social conventions.
    - *Estimation* of action consequences due to current conventions.
  - **Organisational Assistance** consists *on adapting the existing organisation* in order to improve *system's performance* under varying circumstances.
    - i. To observe system evolution
    - ii. To compare it with the organisational goals, and
    - iii. To adapt the organisation trying to improve goal fulfilment.

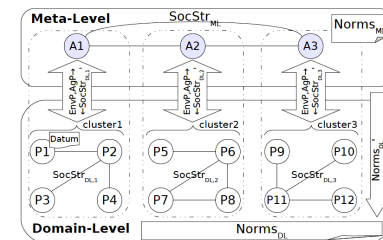
- *Distributed Meta-Level (ML)* that provides assistance to a *Domain-Level (DL)* in charge of domain-specific tasks.
- The two levels communicate through an *interface*.
- Each level has a set of agents with its own organisation
  - DL = <AgDL, OrgDL>
  - ML = <AgML, OrgML>
- Using the *Interface*, ML agents perceive partial information about environmental observable properties (EnvP, date or temperature), and agents' observable properties (AgP, e.g., colour or position).
- a ML agent has partial information about the subset of DL agents it assists.
- DL agents are grouped into clusters according to a domain-specific criterion (e.g., interaction costs).
- A ML agent assists a cluster of DL agents, observes partial information about them, and shares it with other ML agents in order to provide better assistance services.

- P2P sharing networks
  - A set of computers connected to the Internet (peers) share some data.

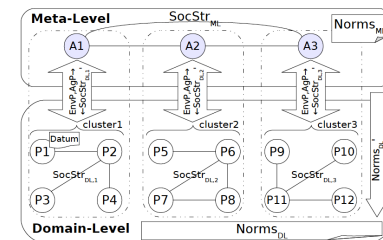




- DL is composed by a set of agents playing the *peer role*.
- DL *Social Structure* determines agents' relationships (sharing data).
- The organisational goal (*Goals*): ***all peers receive the data with the minimal time and network consumptions*** (simulation metrics).
- **Norms:**
  - a peer cannot use more than  $\max_{bw}$  percentage to share data.
  - $\mathbf{normFriends}_{DL}$  a peer cannot simultaneously send the data to more than  $\max_{Friends}$  peers.
- Each ML agent plays the assistant role for a cluster of DL agents (peers).
- It collects information and adapts their local organisation.



- Its decisions are based on:
  - local information about its cluster,
  - aggregated information about other clusters, and
  - the norms at ML
- Ex:
  - Latencies ( $EnvP$ )
  - Which peers have the data ( $AgP$ )
- Norms
  - Limiting the number of peers - in the cluster - to inform about a new peer – in another cluster – having the data



- Simplified version of the *BitTorrent* protocol.

Phase	Level Protocol Messages
initial	ML join<hasDatum>
latency	ML get_latency<peers>, latency<peer><measure> DL lat_req, lat_rpl
social structure	ML contact<peers>
handshake	DL bitfield<hasDatum>
data sharing	DL request, data, cancel ML completed, completed_peer<peer>, has_datum<peer>, all_completed
inactive	DL have
waiting	DL choke, unchoke
norms	ML suggested_bw<value>, suggested_friends<value>, norm_updated<norm_id><new_definition>

**Table 1.** Protocol messages grouped into subsequent phases.

- Peers *join* their cluster by informing its assistant.

- Simplified version of the *BitTorrent* protocol.

Phase	Level Protocol Messages
initial	ML join<hasDatum>
latency	ML get_latency<peers>, latency<peer><measure> DL lat_req, lat_rpl
social structure	ML contact<peers>
handshake	DL bitfield<hasDatum>
data sharing	DL request, data, cancel ML completed, completed_peer<peer>, has_datum<peer>, all_completed
inactive	DL have
waiting	DL choke, unchoke
norms	ML suggested_bw<value>, suggested_friends<value>, norm_updated<norm_id><new_definition>

**Table 1.** Protocol messages grouped into subsequent phases.

- *Assistants need local information:* they initiate latency phases requesting peers to measure their latency with all other peers in their clusters.
- *Assistants propose a social structure* among peers in their clusters.

- Simplified version of the *BitTorrent* protocol.

Phase	Level Protocol Messages
initial	ML join<hasDatum>
latency	ML get_latency<peers>, latency<peer><measure> DL lat_req, lat_rpl
social structure	ML contact<peers>
handshake	DL bitfield<hasDatum>
data sharing	DL request, data, cancel ML completed, completed_peer<peer>, has_datum<peer>, all_completed
inactive	DL have
waiting	DL choke, unchoke
norms	ML suggested_bw<value>, suggested_friends<value>, norm_updated<norm_id><new_definition>

**Table 1.** Protocol messages grouped into subsequent phases.

- Peers perform a *handshake phase* where they introduce themselves to their contacts, and specify whether they have the datum.
- A *data sharing phase* starts – data request and data transmission.
- Otherwise – as soon as one peer receives the datum, it will inform its handshaked peers – the sharing phase is triggered.

- Simplified version of the *BitTorrent* protocol.

Phase	Level Protocol Messages
initial	ML join<hasDatum>
latency	ML get_latency<peers>, latency<peer><measure> DL lat_req, lat_rpl
social structure	ML contact<peers>
handshake	DL bitfield<hasDatum>
data sharing	DL request, data, cancel ML completed, completed_peer<peer>, has_datum<peer>, all_completed
inactive	DL have
waiting	DL choke, unchoke
norms	ML suggested_bw<value>, suggested_friends<value>, norm_updated<norm_id><new_definition>

**Table 1.** Protocol messages grouped into subsequent phases.

- A source peer cannot start a transmission if it is already serving the maximum number of allowed peers.
- In this case transmissions can only be initiated when a previous transmission ends.

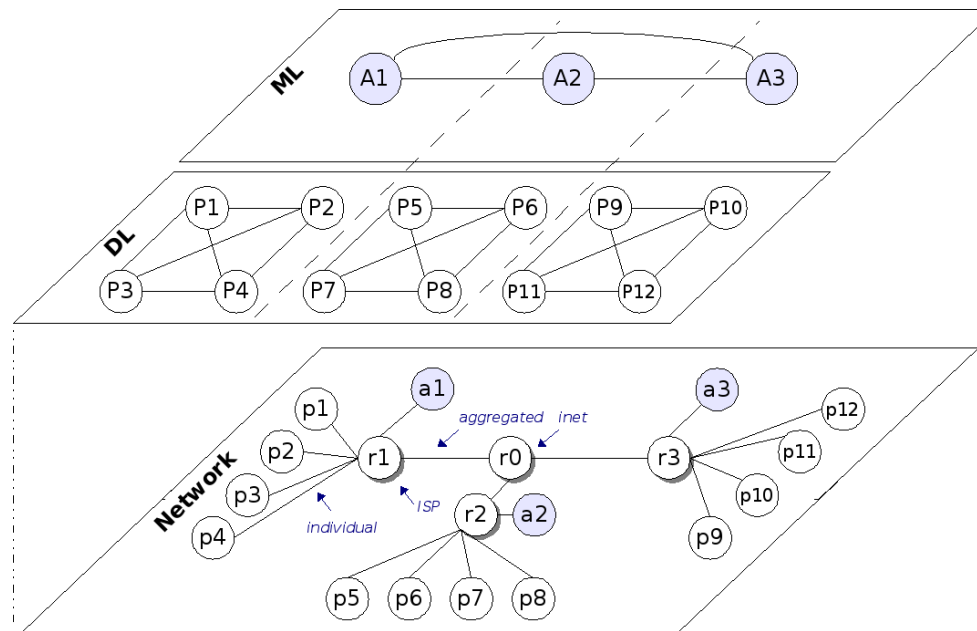
- Simplified version of the *BitTorrent* protocol.

Phase	Level Protocol Messages
initial	ML join<hasDatum>
latency	ML get_latency<peers>, latency<peer><measure> DL lat_req, lat_rpl
social structure	ML contact<peers>
handshake	DL bitfield<hasDatum>
data sharing	DL request, data, cancel ML completed, completed_peer<peer>, has_datum<peer>, all_completed
inactive	DL have
waiting	DL choke, unchoke
norms	ML suggested_bw<value>, suggested_friends<value>, norm_updated<norm_id><new_definition>

**Table 1.** Protocol messages grouped into subsequent phases.

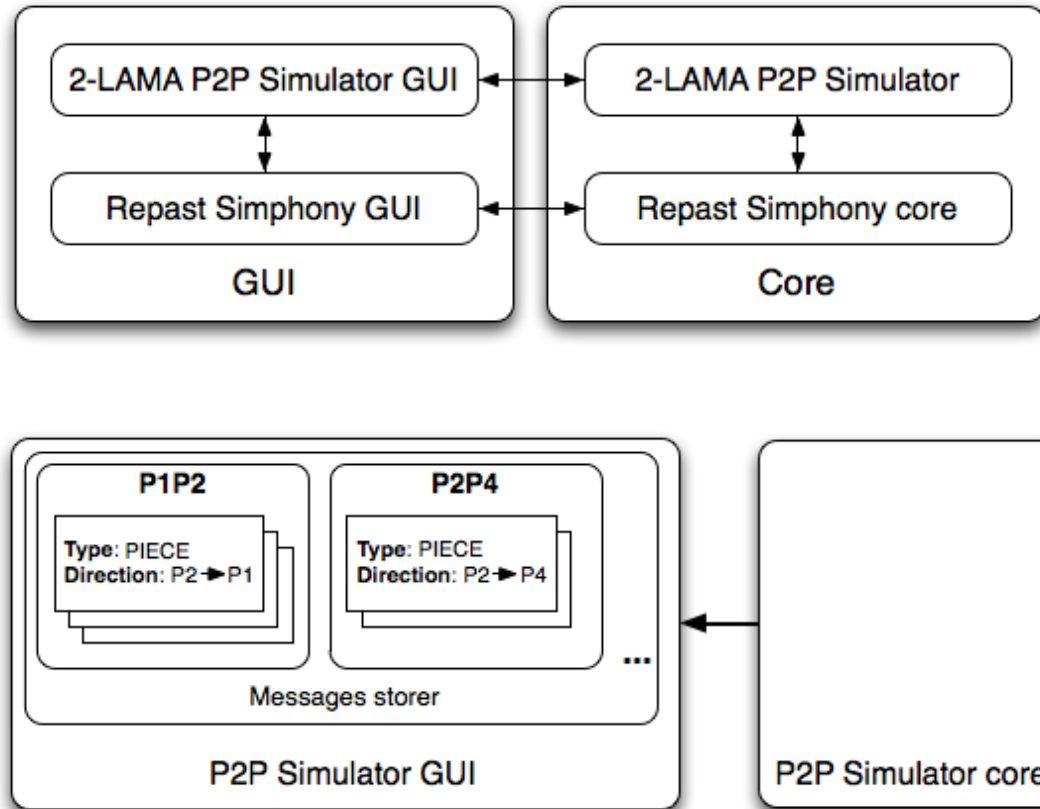
- Norm deliberations and notifications also belong to the protocol.

- A P2P sharing network simulator in Repast Symphony.
- Its architecture allows to:
  - model agents (agent-level), and
  - the transport of messages among them (network-level).

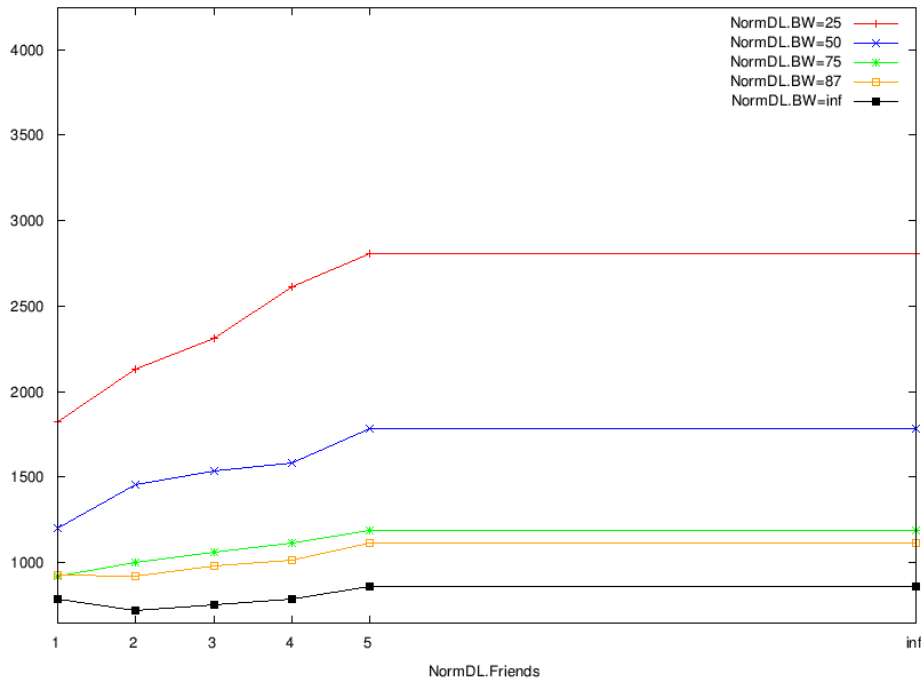




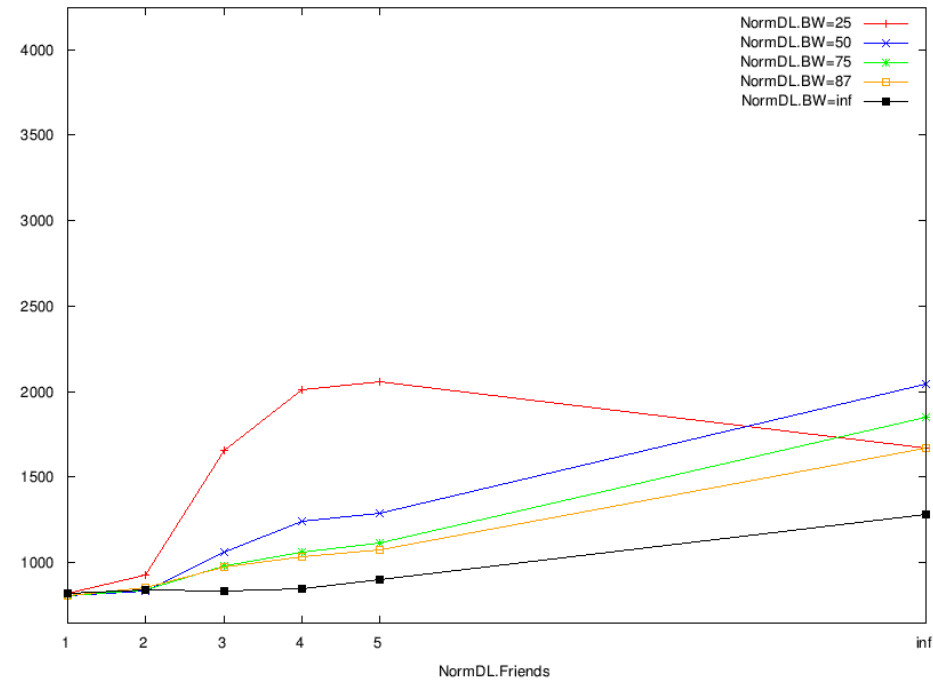
- **P2P module:** to drive the simulation at agent-level
- It provides tools
  - to create state-based agents,
  - to define a problem (number of peers, who has initially datum, etc..),
  - to provide services such as an agents' directory.
- **netsim module**
  - to transport messages among agents,
  - to define different network topologies, and
  - to collect statistical information about network status.
- An agent from the P2P module is attached to a netsim's network adapter, which is charge of actually sending messages.
- **overpast module** that processes the generated logs and extracts relevant information.



[NoAdapt] Time cost (NormML.Has=1)



Time cost (Adap3.4) (NormML.Has=1)



- Time required to share data.
- Network consumed
- Network channel used
- Network channel saturation
- Etc..

- The number of files per node
- The simultaneous connection limit
- The probability of wanting to request a file
- The churn rate

- Dead nodes (red borders)
- Ultra nodes are orange, all others are black
- Light gray lines indicate total known connections
- Green lines indicate current ping requests
- Blue lines indicate current query requests

- MAS that simulates a simple **distributed P2P network for file sharing**.
- Many nodes, which either **download files, host files, or serve as middlemen**.
- Nodes employ local value functions to maximize the speed of each query transaction (Local Decisions).
- The network provides high average query speed across the graph
  - Optimally distributed traffic
  - Low failure rates

- Each agent (node) may exhibit any combination of three core behaviors:
  - Requesting files
  - Hosting (seeding) files
  - Serving as a middleman routing traffic
- The following actions are available to nodes:
  - Ping: Discover peer I
  - Pong: Reply to a ping from peer I
  - Query: Request file f
  - Query Hit: Reply to a query for file f
  - Join: Join Network N
  - Leave: Leave Network N (die)

<https://github.com/michaelhollman/mas-network-simulation>

Jordi Campos Miralles, Maite López-Sánchez, Marc Esteva, Javier Morales  
**A simulator for organisation-centred MAS adaptation in P2P sharing networks.** AAMAS 2010: 1615-1616

Jordi Campos Miralles, Marc Esteva, Maite López-Sánchez, Javier Morales:  
**A simulator for a two layer MAS adaptation in P2P networks.** WAT 2009: 106-117





# Adaptation Mechanisms in P2P Networks

A multi-agent simulation using Repast

**Antonio Bucchiarone**

Fondazione Bruno Kessler, Trento – Italy

[bucchiarone@fbk.eu](mailto:bucchiarone@fbk.eu)

28 October 2019