



# UCL's future work in Evergrow

Peter Van Roy, Raphaël Collet, Kevin Glynn

# Our main topics

- Peer-to-peer computing
  - **P2PS**: a peer-to-peer library
  - **P2PKit**: a service library on top of P2PS
- Distributed languages
  - Mozart and its new distribution implementation

# P2PS, a peer-to-peer library

- Implements a DHT
- Is implemented in Mozart, using the transparent distribution
  - ✓ The implementation is made in a high-level language
  - ✗ Currently a bit limiting, because the Mozart connections do not always reflect the P2P connections
- Is being improved for more robustness (reliable communication, more stable routing, etc.)

# P2PKit, a service architecture

- Extends the P2PS functionality with **user services**
- Available services:
  - Monitoring (ControlGUI)
  - Failure detection
  - Replicated data storage (no symmetric replication yet)

# More services to come

- Service directory
  - Holds meta-data and implementation for public services running in the network
  - Could become a service broker
- Decentralized transaction manager
  - Implements distributed transactions
  - Could become a decentralized DBMS

# More services to come

- Selfman project
  - Component-based architecture with hooks and reflective capabilities
  - To support self-managing, self-optimizing, upgradable decentralized services and applications
- Application-level encryption and certificates
- Group communication

# Mozart, a distributed language

- We are reimplementing the distribution layer of Mozart on top of **DSS** (Distribution SubSystem, SICS)
- Will improve the tradeoff between network transparency and network awareness:
  - Annotations, for choosing protocols, gc scheme...
  - Fault reflection that does **not** break transparency
  - **Reflective routing**, to use a P2P network as the underlying transport layer

# Reflective routing

- Make the distribution layer of Mozart use the peer routing of P2PS!
  - **Mozart processes connectivity coincides with user-level peer-to-peer network**
- The DSS must be given “virtual connections” between sites:
  - Use TCP when sites are neighbors in P2PS
  - Use P2PS routing otherwise

# Conclusion

Enable a high-level component architecture

- **P2PKit** enables decentralized lightweight services
- **P2PS** enables peer-to-peer networking
- **The new Mozart distribution** enables very high-level programming, by keeping network transparency