

Policy system of the Socket Intents prototype

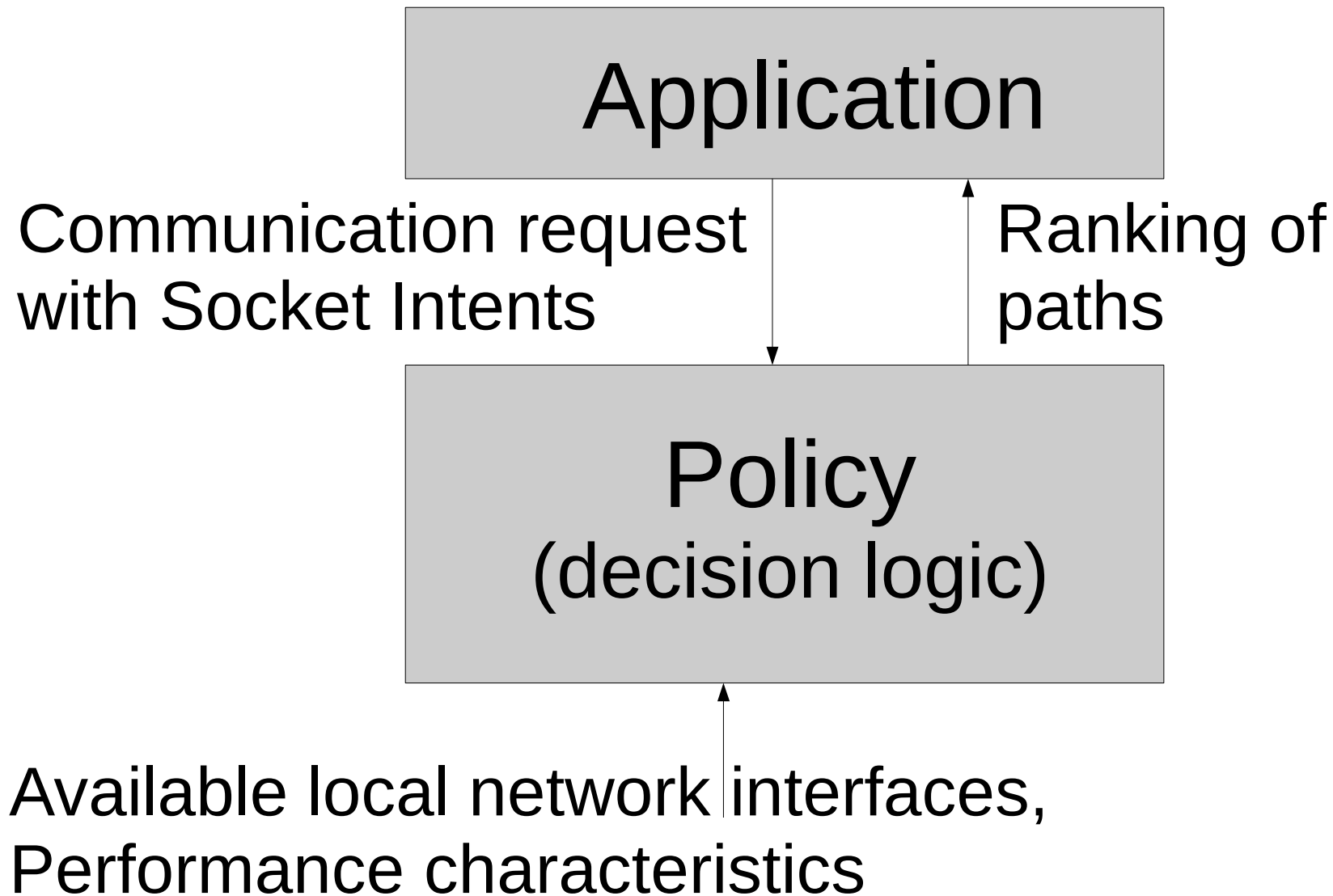
Theresa Enhardt
theresa@inet.tu-berlin.de

Philipp S. Tiesel
philipp@inet.tu-berlin.de

Socket Intents

- Focus: Improve application performance
- Scenario: Multiple permissible paths available
- Select „the better“ path by selecting a local network interface
- Based on heuristics, e.g., estimate download time

Socket Intents Policies



Policy inputs (1/2)

- Socket Intents:
 - What to optimize for
 - Application tells us what it wants, prefers, or expects
 - e.g., "Avoid costly interfaces"
 - e.g., "File Size to receive: 100 MB"

Policy inputs (2/2)

- Path characteristics
 - Per-interface
 - Gathered periodically by observing local traffic
 - Aggregated within the prototype

Policy decisions

- Select a path (via local network interface)
- Select between remote endpoints (via resolved IP address)
- Choose to reuse a socket from a socket pool
- Set socket options

Per-message Path Selection

- Not trivial for stream-based transport protocols
- Implemented via connection pools
 - Policy selects which connection to open or re-use, thus selects path
- How to assign messages to paths?
 - Multiple TCP connections: Same endpoint, different interfaces
 - Multi-streaming is beneficial
 - Multipath QUIC?

Policy implementation

- „Multi Access Manager“
 - Daemon running on the host
 - Collects path characteristics
 - Loads policy as a module
 - Can exchange policy at run-time
- <https://github.com/fg-inet/socket-intents>

Outlook

- Explore performance benefits
 - Which Intents and metrics help improve performance the most?
 - How much do they improve?
- Formalize Socket Intents policies using a Domain Specific Language