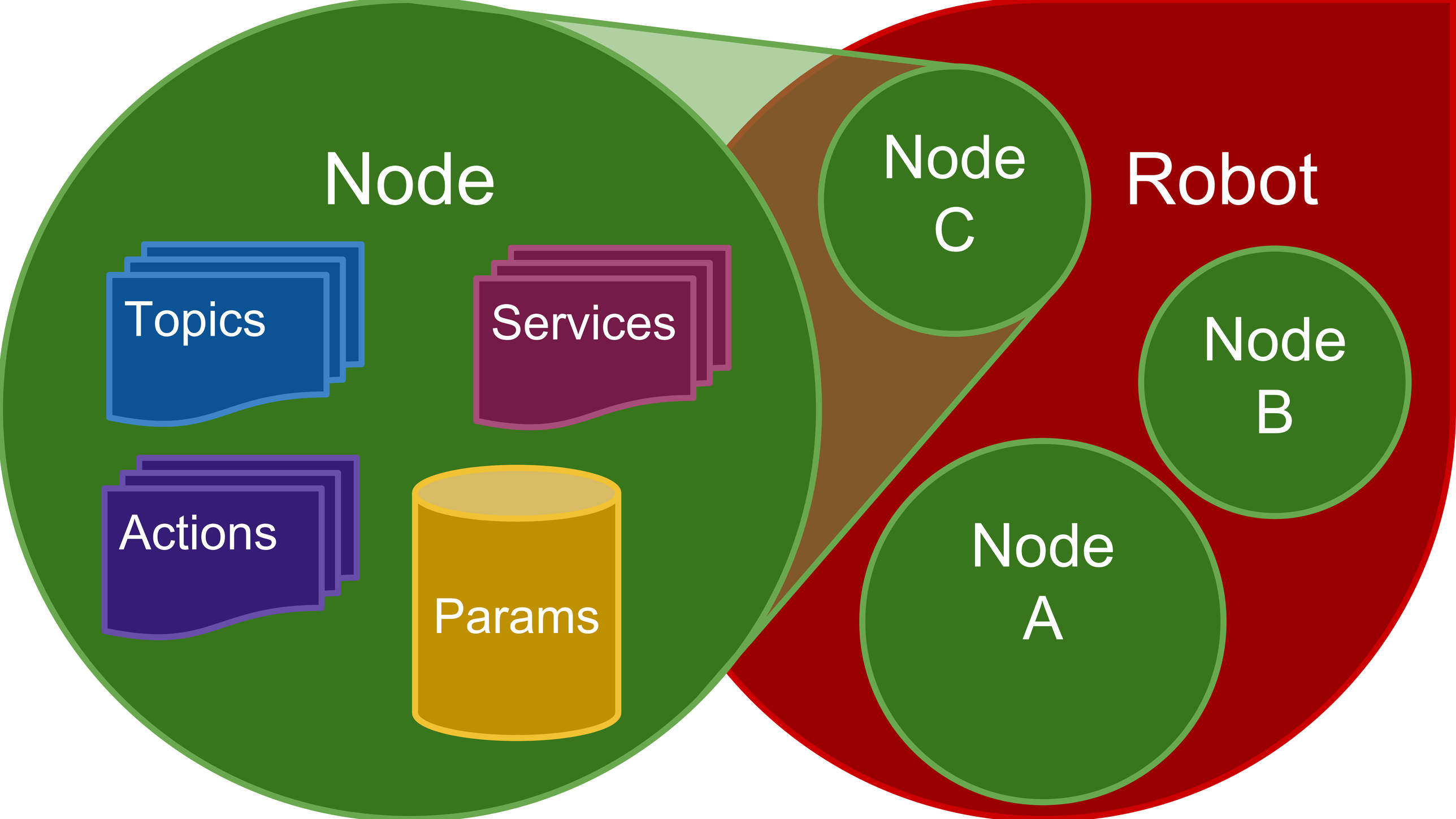
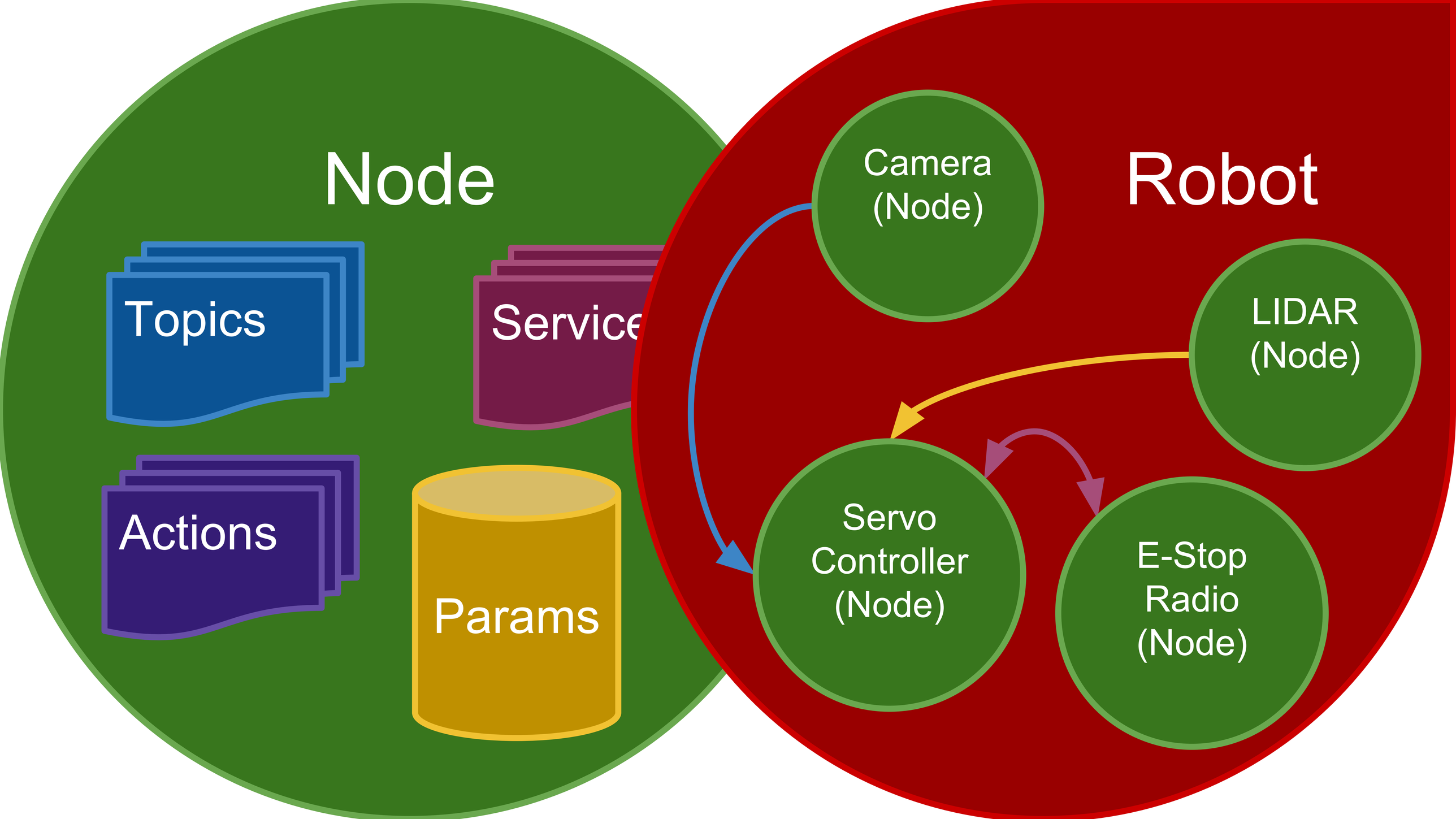
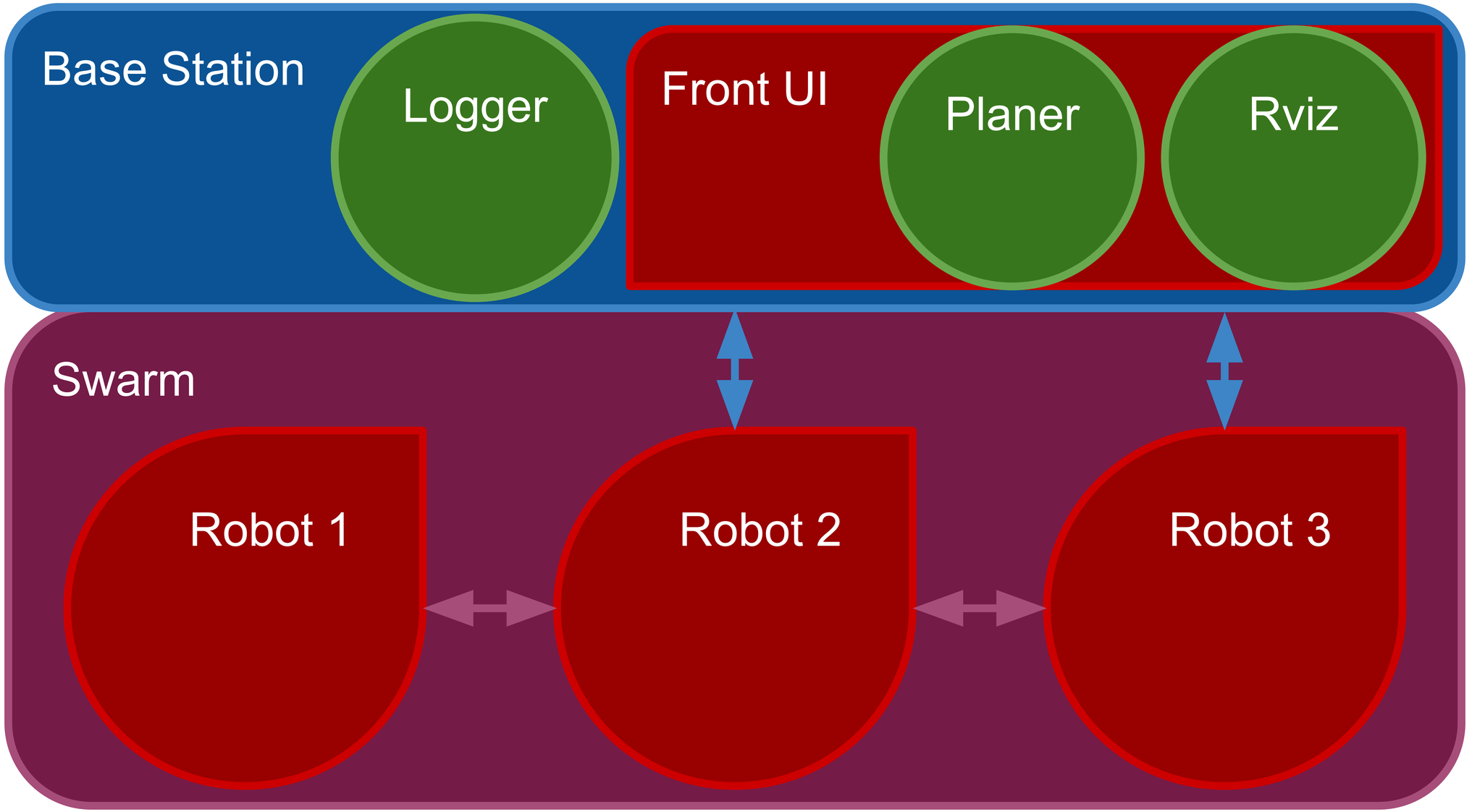
Procedurally Provisioned Access Control for Robotic Systems

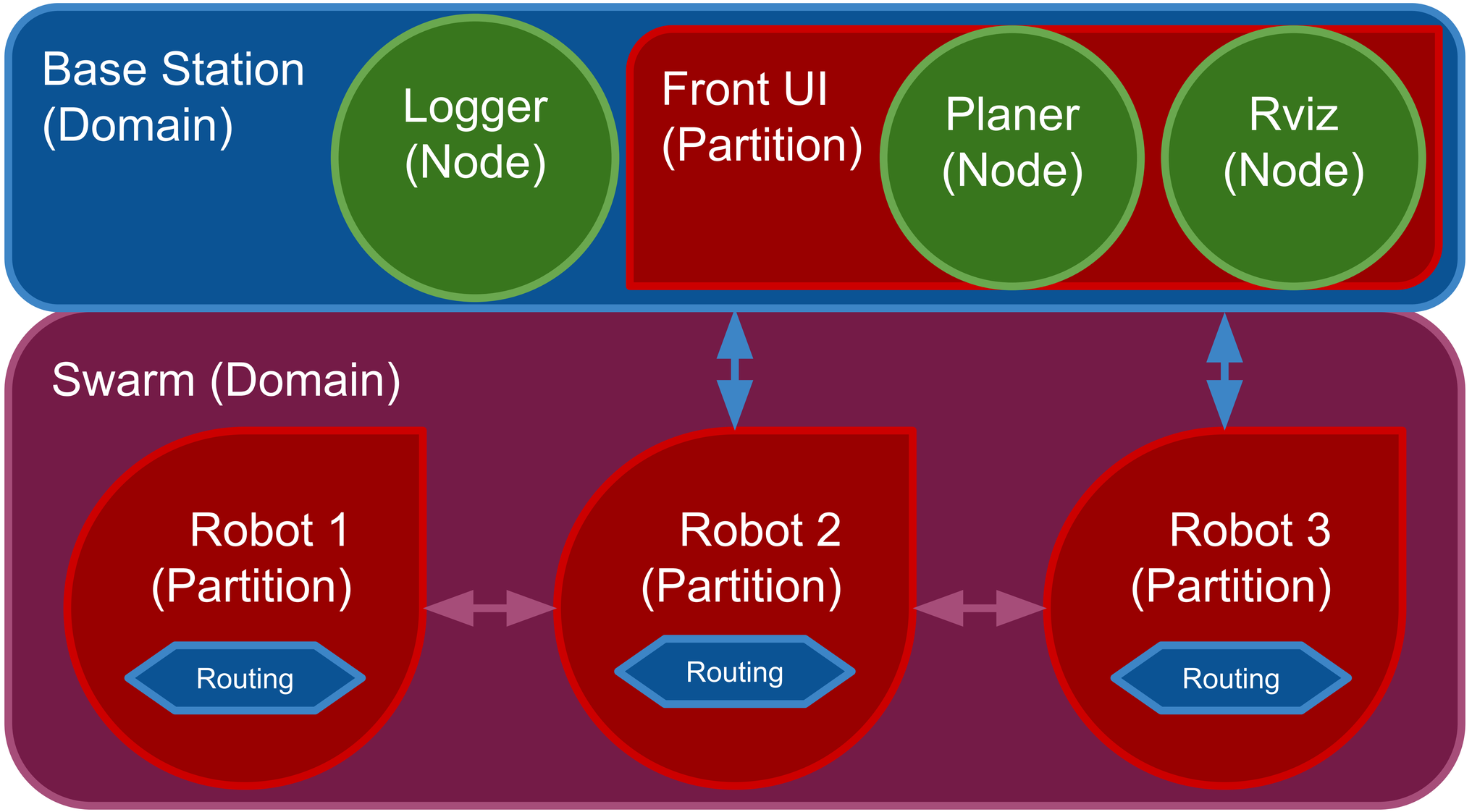
Ruffin White[1], Gianluca Caiazza[2],
Agostino Cortesi[2], Henrik I. Christensen[1]

[1]Contextual Robotics Institute UC San Diego
[2]Ca' Foscari University of Venice


Node
Topics
Services
Actions
Params
Robot
Node C
Node B
Node A


Node
Topics
Service
Actions
Params
Robot
Camera
(Node)
LIDAR
(Node)
Servo
Controller
(Node)
E-Stop
Radio
(Node)


Base Station
Logger
Front UI
Planer
Rviz
Swarm
Robot 1
Robot 2
Robot 3


Base Station
(Domain)
Logger
(Node)
Front UI
(Partition)
Planer
(Node)
Rviz
(Node)
Swarm (Domain)
Robot 1
(Partition)
Routing
Robot 2
(Partition)
Routing
Robot 3
(Partition)
Routing

ComArmor

Profiles are Attached to subjects via URI (*Namespace*)

Attachment is an expression used to match a URI

Profiles are composed of object access Rules or nested profiles

Rules specify object type, attachment, and permissions the policy allows or denies

```
<profiles xmlns:xi="http://www.w3.org/2001/XInclude">
    <xi:include href="tunables/global.xml" parse="xml"/>
    <profile name="My Talker Profile">
        <attachment>/talker</attachment>
        <xi:include href="tunables/node.xml" parse="xml"/>
        <topic qualifier="ALLOW">
            <attachment>/chatter</attachment>
            <permissions>
                <publish/>
            </permissions>
        </topic>
    </profile>
    <profile name="My Listener Profile">
        <attachment>/listener</attachment>
        <xi:include href="tunables/node.xml" parse="xml"/>
        <topic qualifier="ALLOW">
            <attachment>/chatter</attachment>
            <permissions>
                <subscribe/>
            </permissions>
        </topic>
    </profile>
</profiles>
```

ComArmor

Profiles are Attached to subjects via URI (*Namespace*)

Attachment is an expression used to match a URI

Profiles are composed of object access Rules or nested profiles

Rules specify object type, attachment, and permissions the policy allows or denies

```
<profiles xmlns:xi="http://www.w3.org/2001/XInclude">
    <xi:include href="tunables/global.xml" parse="xml"/>
    <profile name="My Talker Profile">
        <attachment>/talker</attachment>
        <xi:include href="tunables/node.xml" parse="xml"/>
        <topic qualifier="ALLOW">
            <attachment>/chatter</attachment>
            <permissions>
                <publish/>
            </permissions>
        </topic>
    </profile>
    <profile name="My Listener Profile">
        <attachment>/listener</attachment>
        <xi:include href="tunables/node.xml" parse="xml"/>
        <topic qualifier="ALLOW">
            <attachment>/chatter</attachment>
            <permissions>
                <subscribe/>
            </permissions>
        </topic>
    </profile>
</profiles>
```

profile

rule

nested includes

allowed permission

Keymint: automated cryptographic build tool

comarmor.d/* (example.xml)

```
Profile:
  Attachment: /foo/*/wheatley
    #include <tunables/node>
    param /use_sim_time r,
    topic /chatter{,/**} p,
    deny topic /chatter/foo p,
    deny topic /*/e-stop{,/**} p,
    service /wheatley/get_loggers x,
    service /wheatley/set_logger_level x,
```

keystore.cnf

```
  Identity CA:
   Issuer:
    Aperture Sci
   Hash: SHA256
   Type: RSA
   Size: 4096
   Valid: ~52k AD
...
```

```
$ tree keymint_ws/
keymint_ws/
└── profile
    ├── comarmor.d
    │   ├── example.xml
    │   └── ...
    └── keystore.cnf
```

Keymint: automated cryptographic build tool

comarmor.d/* (example.xml)

```
Profile:
  Attachment: /foo/*/wheatley
    #include <tunables/node>
    param /use_sim_time r,
    topic /chatter{,/**} p,
    deny topic /chatter/foo p,
    deny topic /*/e-stop{,/**} p,
    service /wheatley/get_loggers x,
    service /wheatley/set_logger_level x,
```

keymint_package.xml

```
Format:
    keymint_ros2_dds
...
```

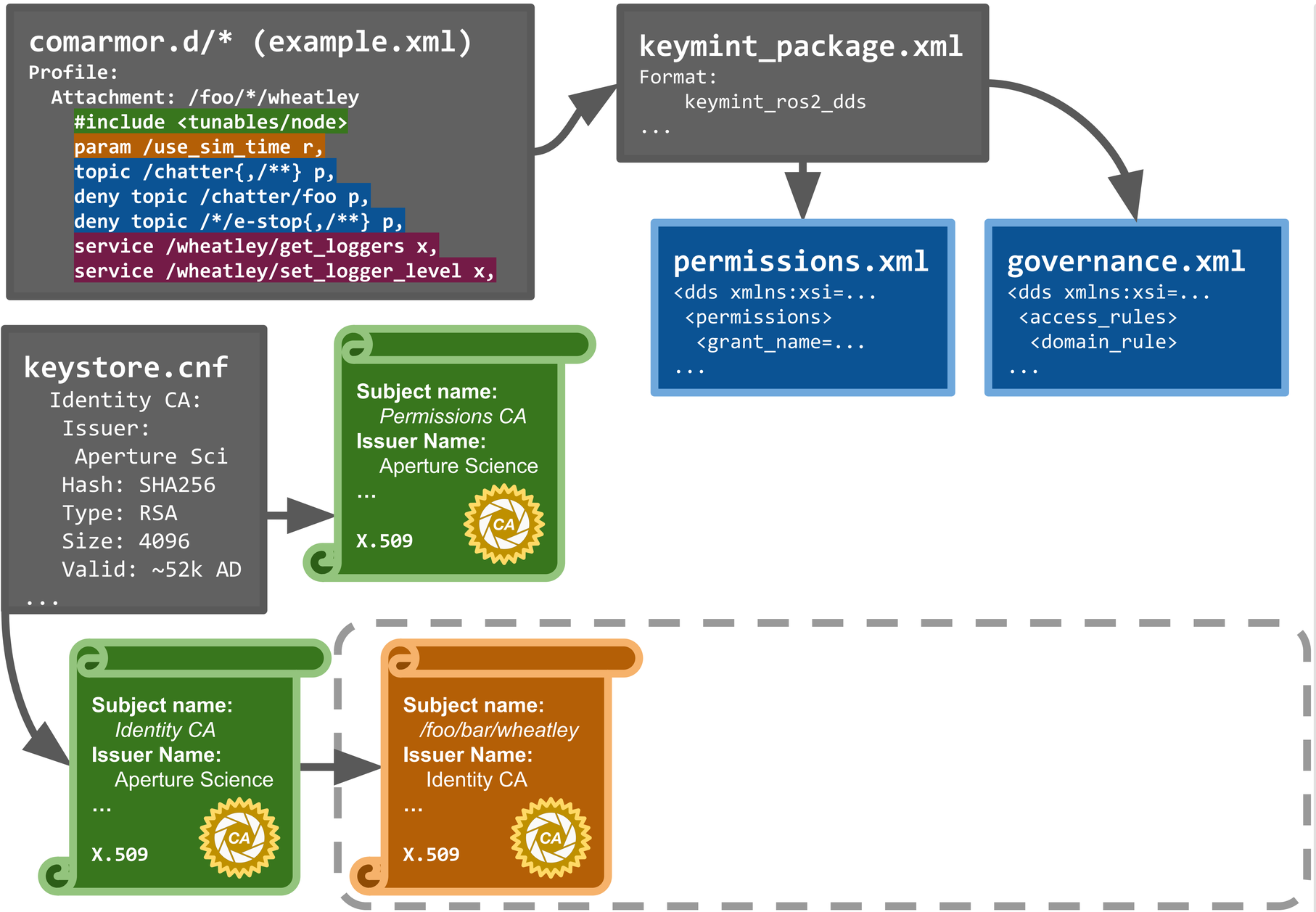
keystore.cnf

```
  Identity CA:
   Issuer:
    Aperture Sci
   Hash: SHA256
   Type: RSA
   Size: 4096
   Valid: ~52k AD
...
```


```
$ keymint create “/foo/bar/wheatley”


$ tree keymint_ws/
keymint_ws/
├── src/foo/bar
│   └── wheatley
│       └── keymint_package.xml
├── private
│   ├── identity.key.pem
│   └── permissions.key.pem
├── profile
│   ├── comarmor.d
│   │   ├── example.xml
│   │   └── ...
│   └── keystore.cnf
└── public
    ├── identity.cert.pem
    └── permissions.cert.pem
```

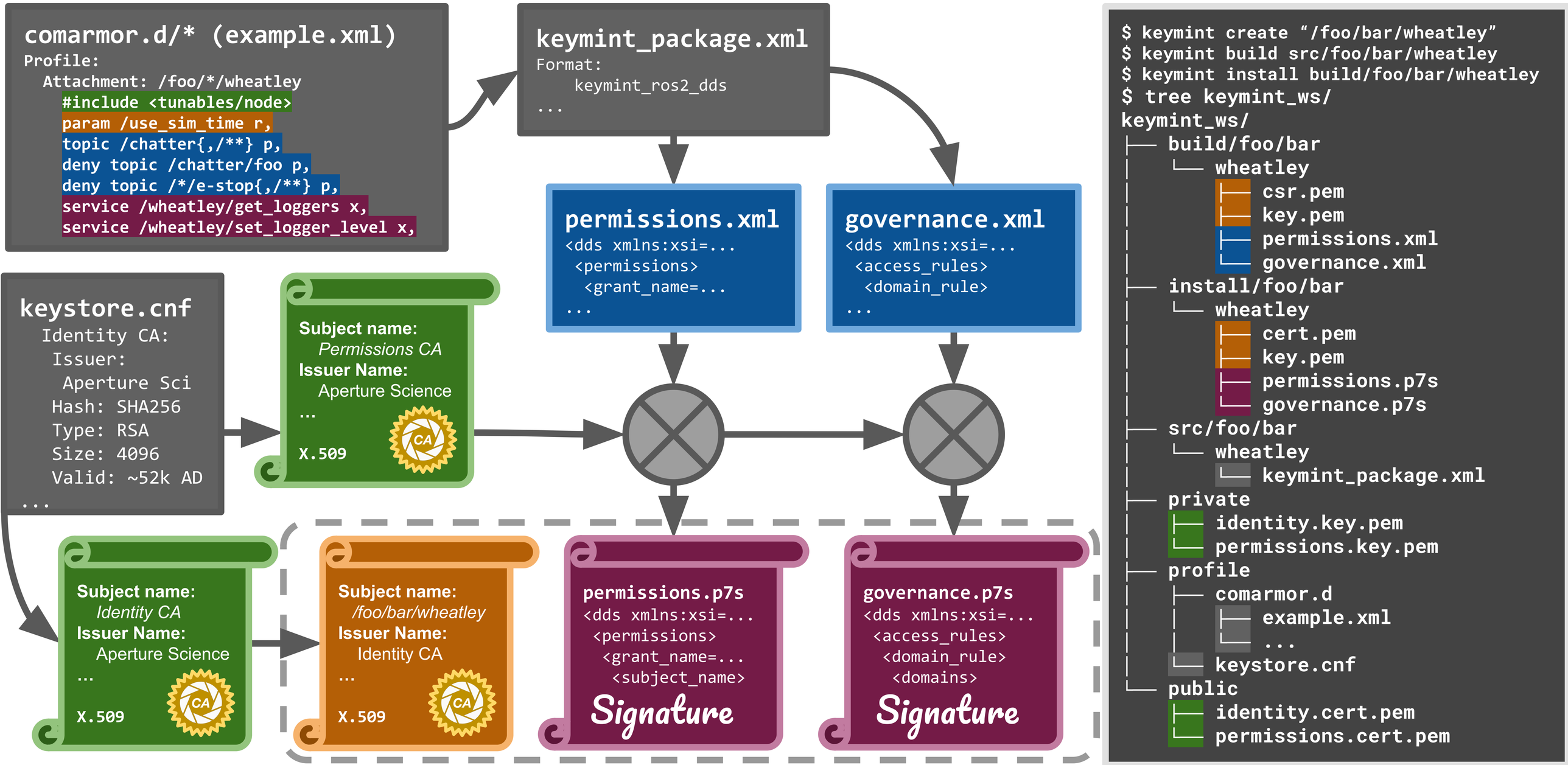
Keymint: automated cryptographic build tool




```
$ keymint create “/foo/bar/wheatley”
$ keymint build src/foo/bar/wheatley

$ tree keymint_ws/
keymint_ws/
├── build/foo/bar
│   └── wheatley
│       ├── csr.pem
│       ├── key.pem
│       ├── permissions.xml
│       └── governance.xml
├── src/foo/bar
│   └── wheatley
│       └── keymint_package.xml
├── private
│   ├── identity.key.pem
│   └── permissions.key.pem
├── profile
│   ├── comarmor.d
│   │   ├── example.xml
│   │   └── ...
│   └── keystore.cnf
└── public
    ├── identity.cert.pem
    └── permissions.cert.pem
```

Keymint: automated cryptographic build tool



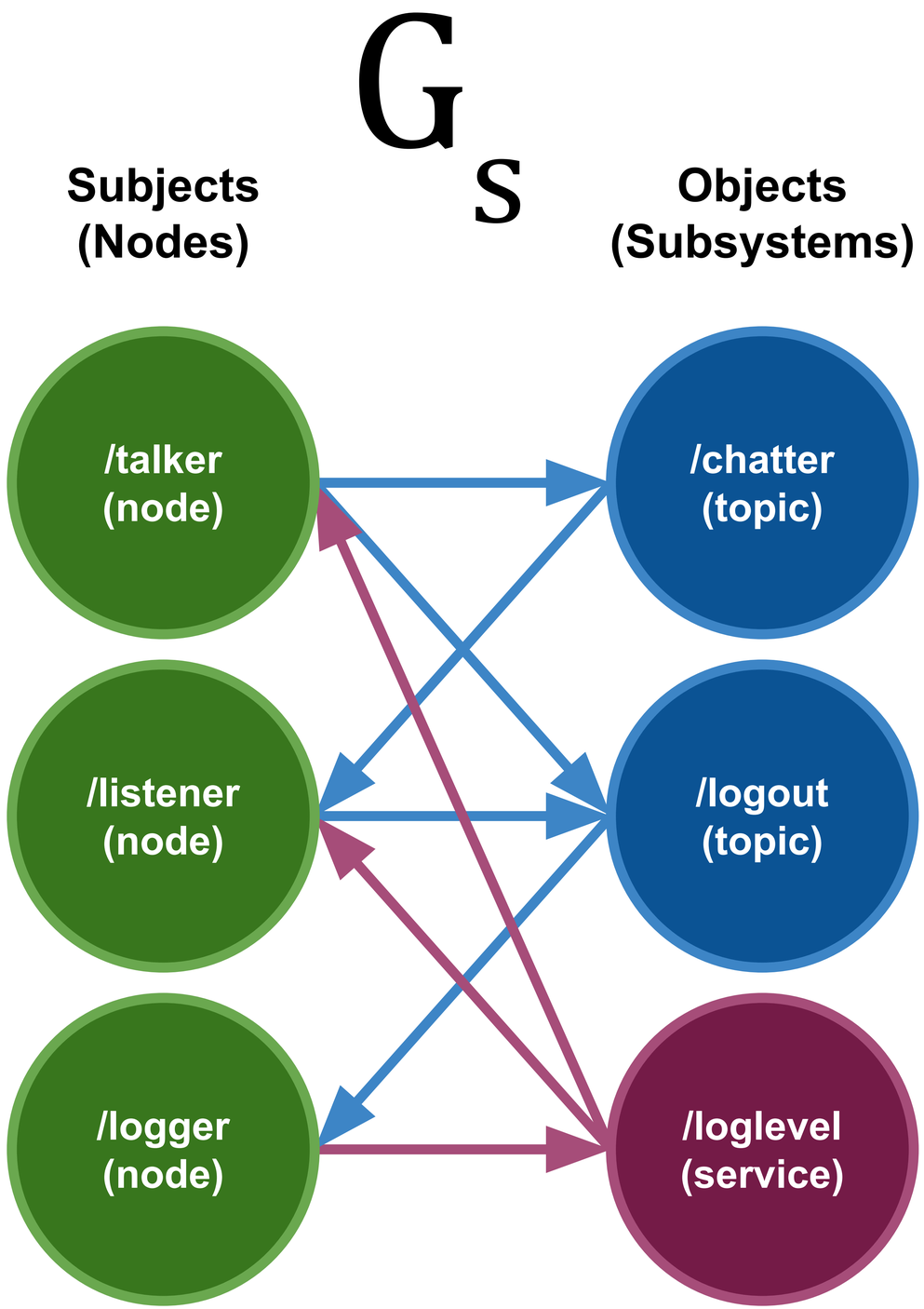

```
$ keymint create “/foo/bar/wheatley”
$ keymint build src/foo/bar/wheatley
$ keymint install build/foo/bar/wheatley
$ tree keymint_ws/
keymint_ws/
├── build/foo/bar
│   └── wheatley
│       ├── csr.pem
│       ├── key.pem
│       ├── permissions.xml
│       └── governance.xml
├── install/foo/bar
│   └── wheatley
│       ├── cert.pem
│       ├── key.pem
│       ├── permissions.p7s
│       └── governance.p7s
├── src/foo/bar
│   └── wheatley
│       └── keymint_package.xml
├── private
│   ├── identity.key.pem
│   └── permissions.key.pem
├── profile
│   ├── comarmor.d
│   │   ├── example.xml
│   │   └── ...
│   └── keystore.cnf
└── public
    ├── identity.cert.pem
    └── permissions.cert.pem
```

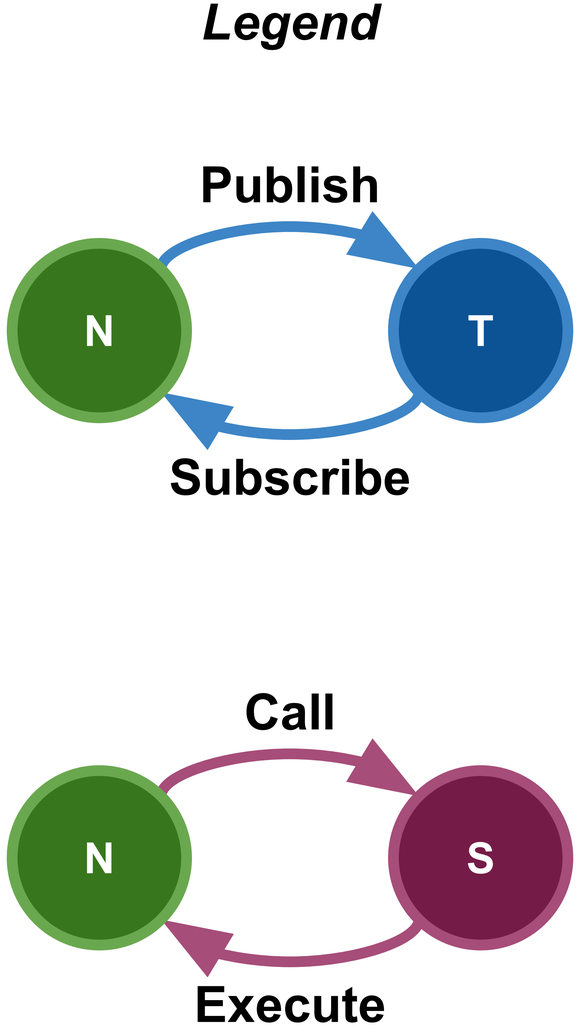
Experiment

Semantic representation of sub-systems modeled as set of bipartite graphs

Subject permission duality visualized by directional edges

Fully Connected bigraph used to verify transport policy compliance





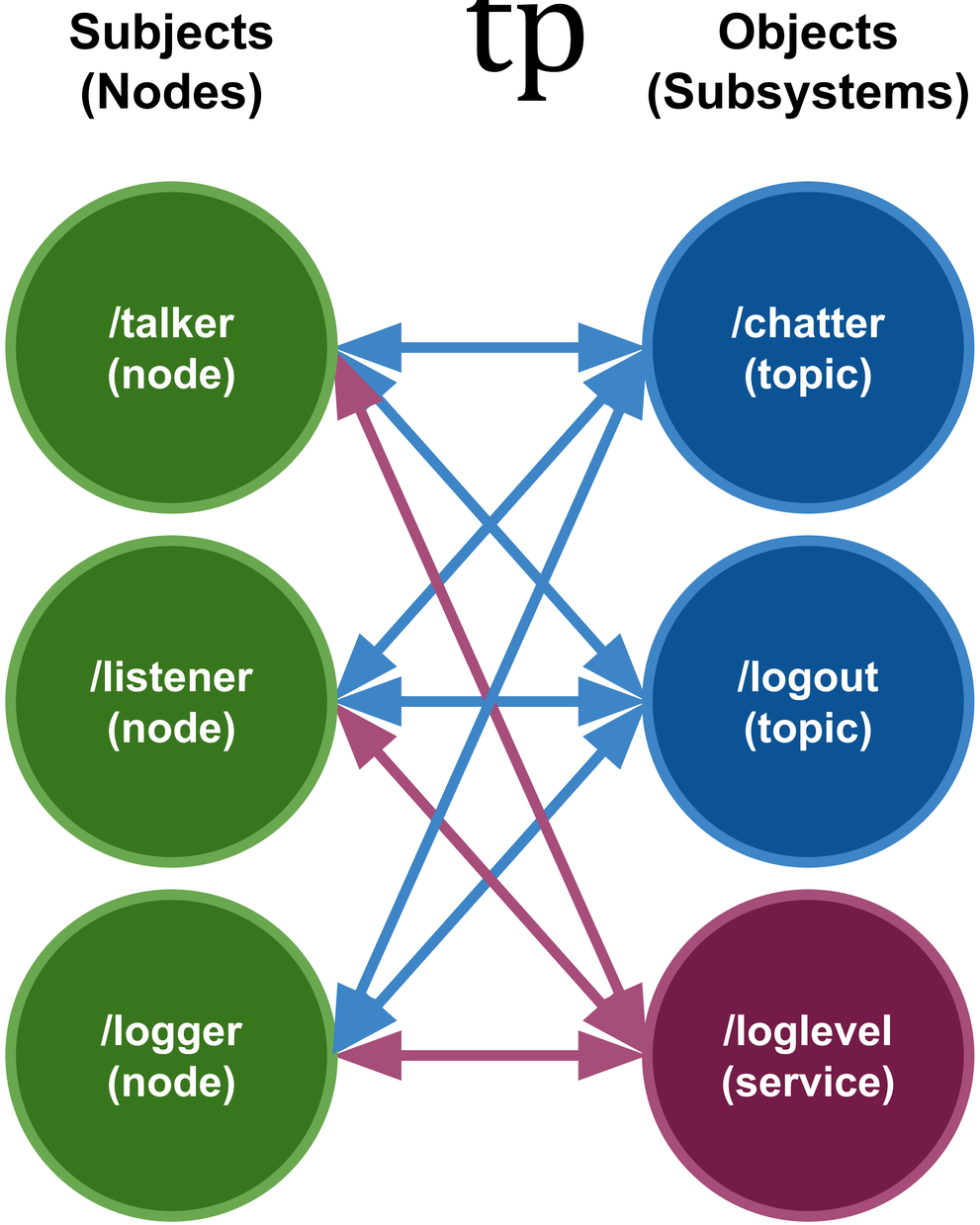
Experiment

Semantic representation of sub-systems modeled as set of bipartite graphs

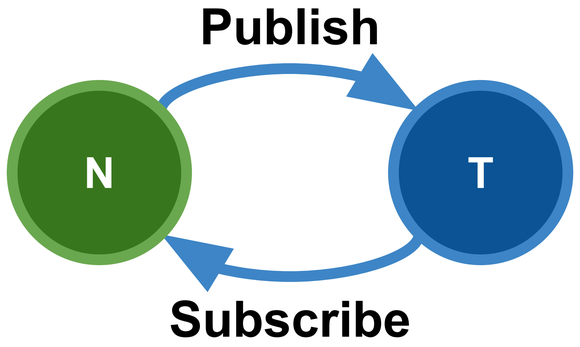
Subject permission duality visualized by directional edges

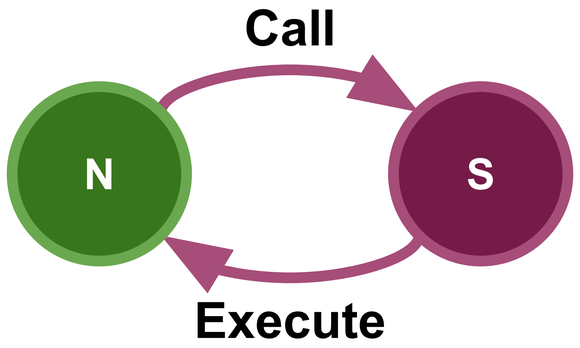
Fully Connected bigraph used to verify transport policy compliance

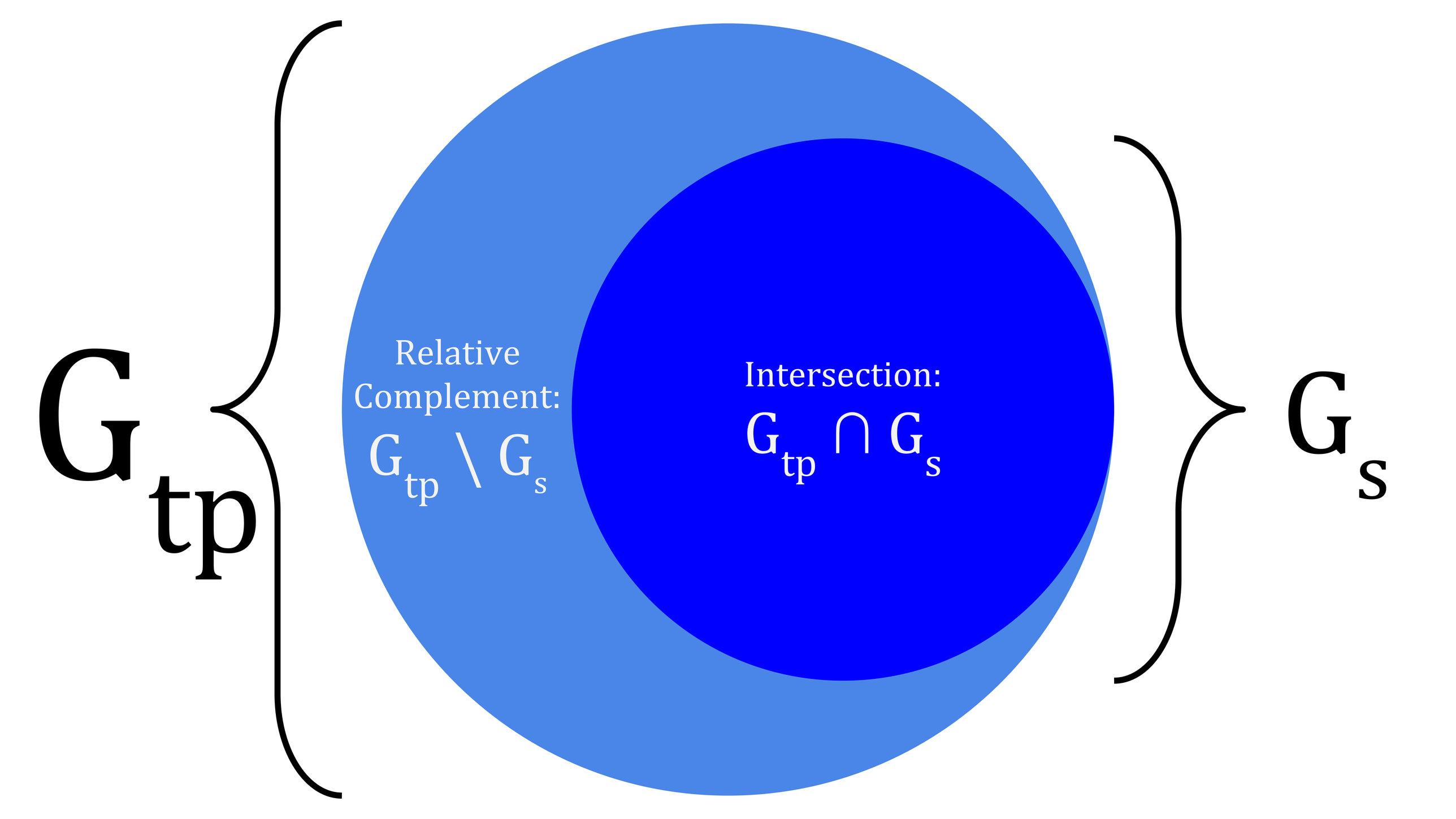
G_{tp}

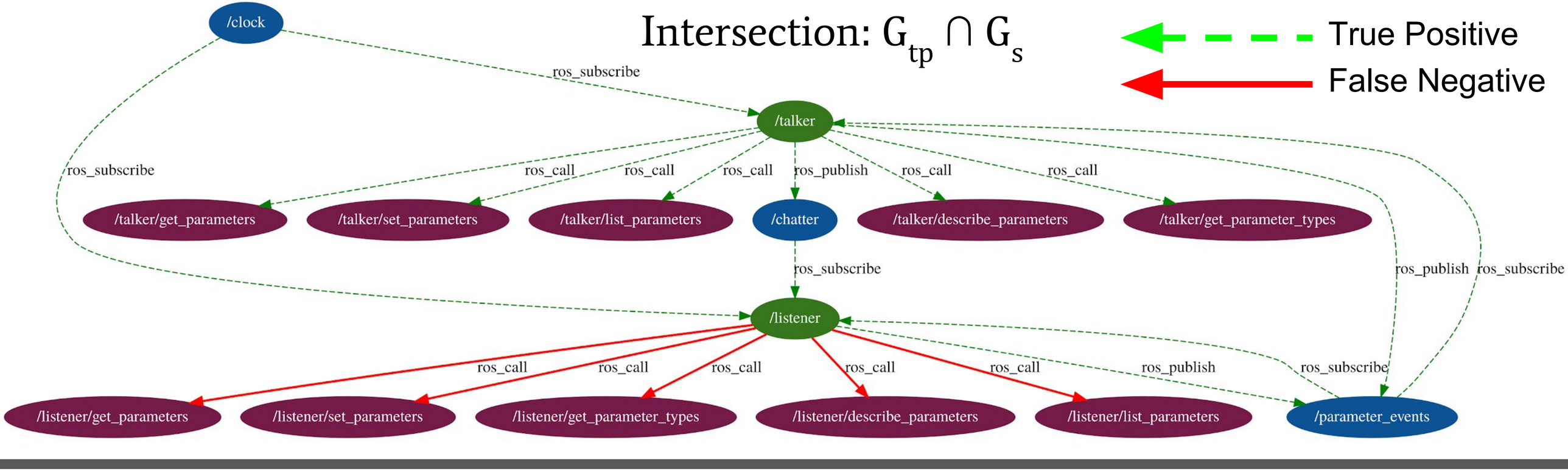


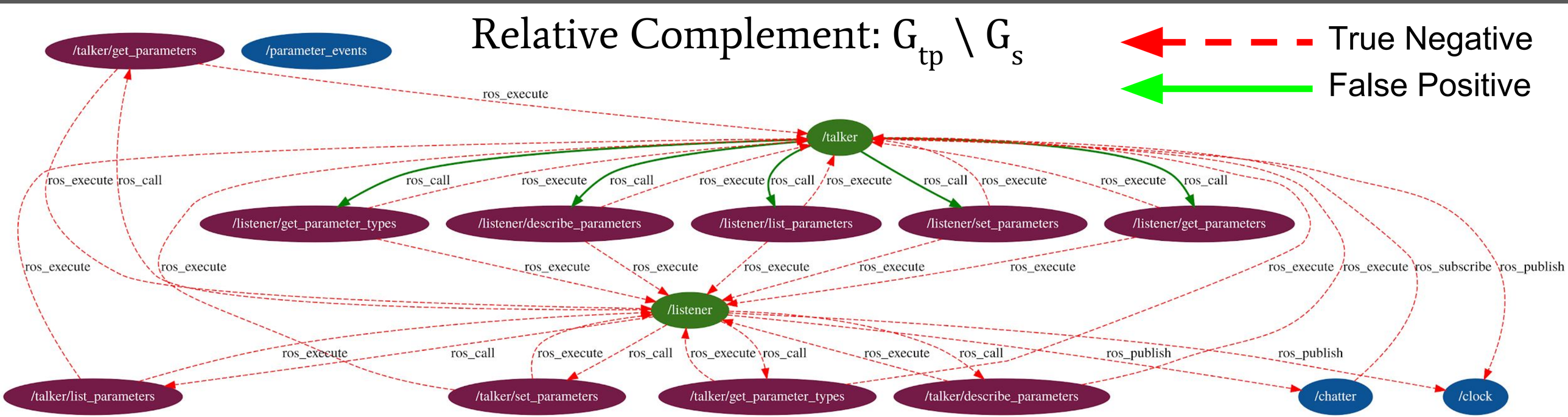
Role Permissions
(Directional Edges)
Legend

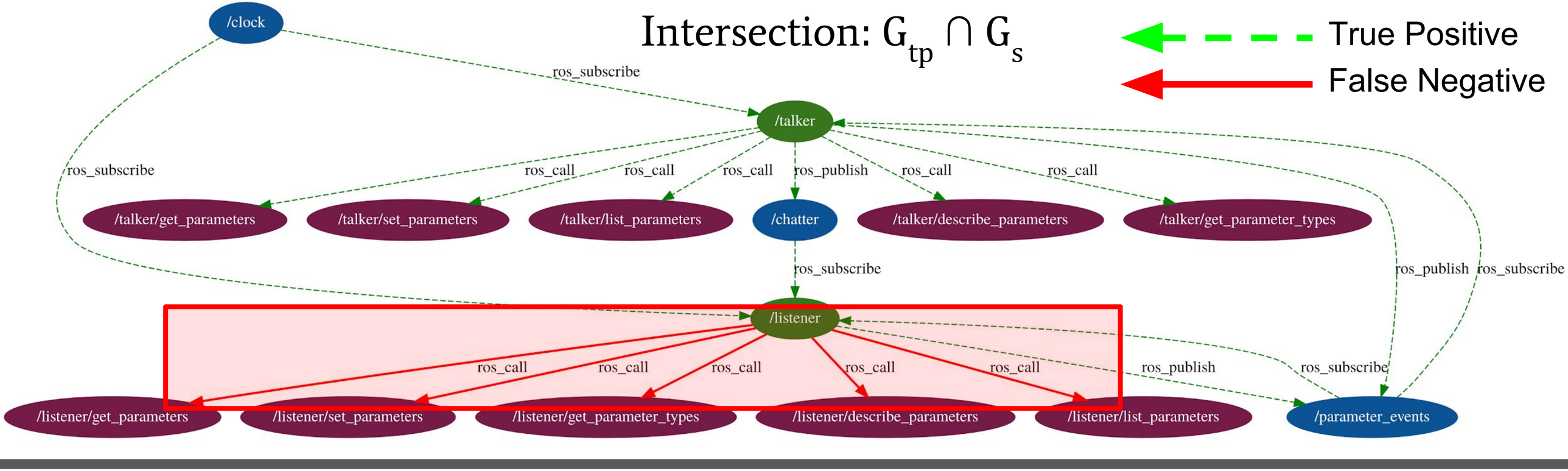


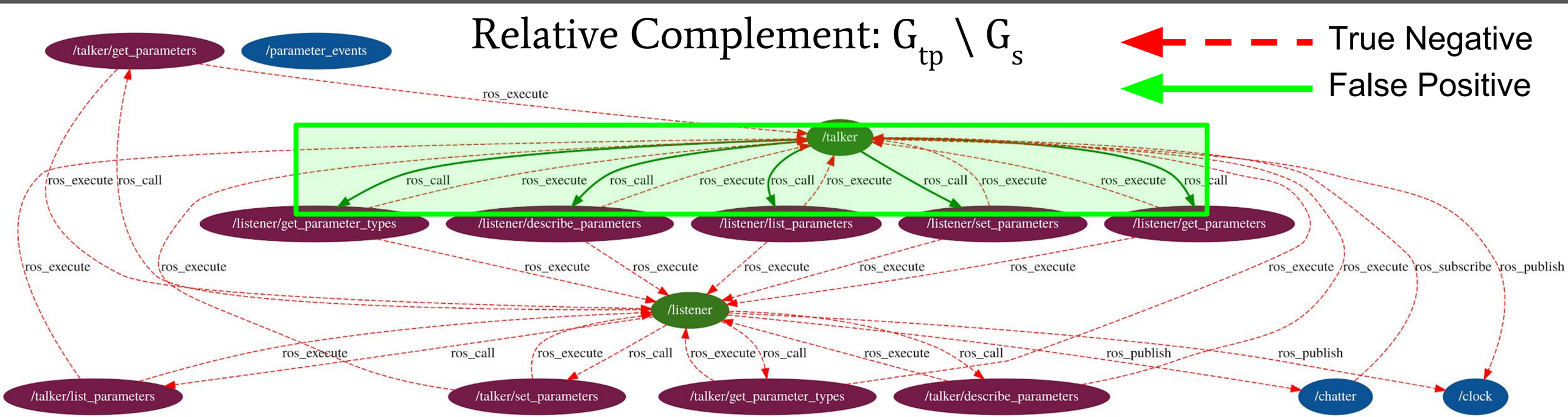



G_{tp}
Relative Complement:
$G_{tp} \setminus G_s$
Intersection:
$G_{tp} \cap G_s$
G_s


Intersection: $G_{tp} \cap G_s$
True Positive
False Negative
/clock
ros_subscribe
/talker
ros_subscribe
ros_call
ros_call
ros_call
ros_publish
ros_call
ros_call
/talker/get_parameters
/talker/set_parameters
/talker/list_parameters
/chatter
/talker/describe_parameters
/talker/get_parameter_types
ros_subscribe
ros_publish
ros_subscribe
/listener
ros_call
ros_call
ros_call
ros_call
ros_call
ros_publish
ros_subscribe
/listener/get_parameters
/listener/set_parameters
/listener/get_parameter_types
/listener/describe_parameters
/listener/list_parameters
/parameter_events


Relative Complement: $G_{tp} \setminus G_s$
True Negative
False Positive
/talker/get_parameters
/parameter_events
ros_execute
/talker
ros_execute
ros_call
ros_call
ros_execute
ros_call
ros_execute
ros_call
ros_execute
ros_call
ros_execute
ros_call
ros_execute
/listener/get_parameter_types
/listener/describe_parameters
/listener/list_parameters
/listener/set_parameters
/listener/get_parameters
ros_execute
ros_execute
ros_execute
ros_execute
ros_execute
ros_execute
ros_execute
ros_execute
ros_subscribe
ros_publish
/listener
ros_execute
ros_call
ros_execute
ros_call
ros_execute
ros_call
ros_execute
ros_call
ros_publish
ros_publish
/talker/list_parameters
/talker/set_parameters
/talker/get_parameter_types
/talker/describe_parameters
/chatter
/clock


Intersection: $G_{tp} \cap G_s$
True Positive
False Negative
/clock
ros_subscribe
/talker
ros_subscribe
ros_call
ros_call
ros_call
ros_publish
ros_call
ros_call
/talker/get_parameters
/talker/set_parameters
/talker/list_parameters
/chatter
/talker/describe_parameters
/talker/get_parameter_types
ros_subscribe
ros_publish
ros_subscribe
/listener
ros_call
ros_call
ros_call
ros_call
ros_call
ros_publish
ros_subscribe
/listener/get_parameters
/listener/set_parameters
/listener/get_parameter_types
/listener/describe_parameters
/listener/list_parameters
/parameter_events


Relative Complement: $G_{tp} \setminus G_s$
True Negative
False Positive
/talker/get_parameters
/parameter_events
ros_execute
/talker
ros_execute
ros_call
ros_call
ros_execute
ros_call
ros_execute
ros_call
ros_execute
ros_call
ros_execute
ros_execute
ros_call
/listener/get_parameter_types
/listener/describe_parameters
/listener/list_parameters
/listener/set_parameters
/listener/get_parameters
ros_execute
ros_execute
ros_execute
ros_execute
ros_execute
ros_execute
ros_execute
ros_execute
ros_subscribe
ros_publish
/listener
ros_execute
ros_call
ros_execute
ros_call
ros_execute
ros_call
ros_execute
ros_call
ros_publish
ros_publish
/talker/list_parameters
/talker/set_parameters
/talker/get_parameter_types
/talker/describe_parameters
/chatter
/clock