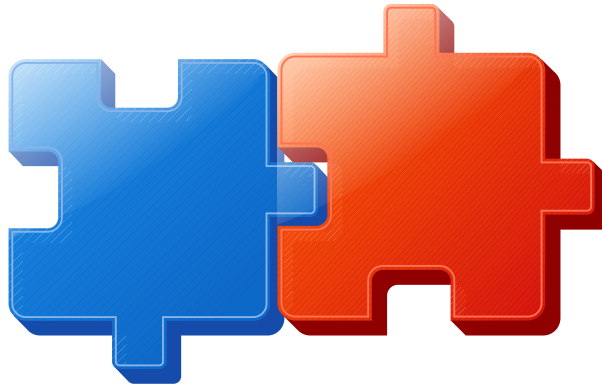


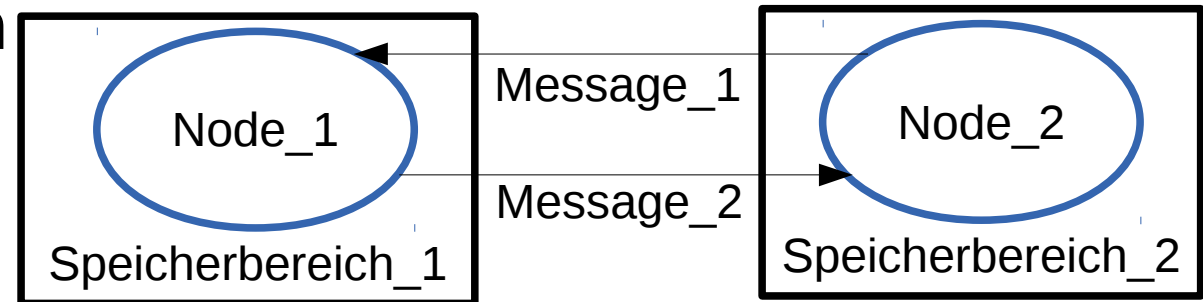
Martin Seidel



ROS nodelets

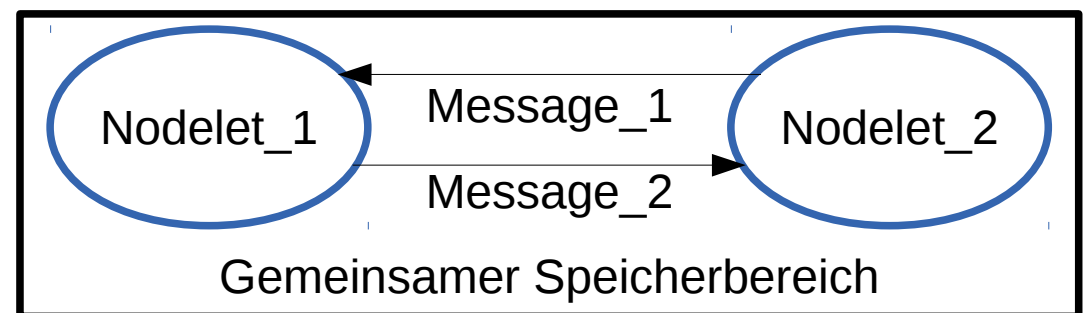
- 2 Nodes:

- Nachrichteninhalt wird kopiert zwischen den Speicherbereichen
- Std. Pub-Sub



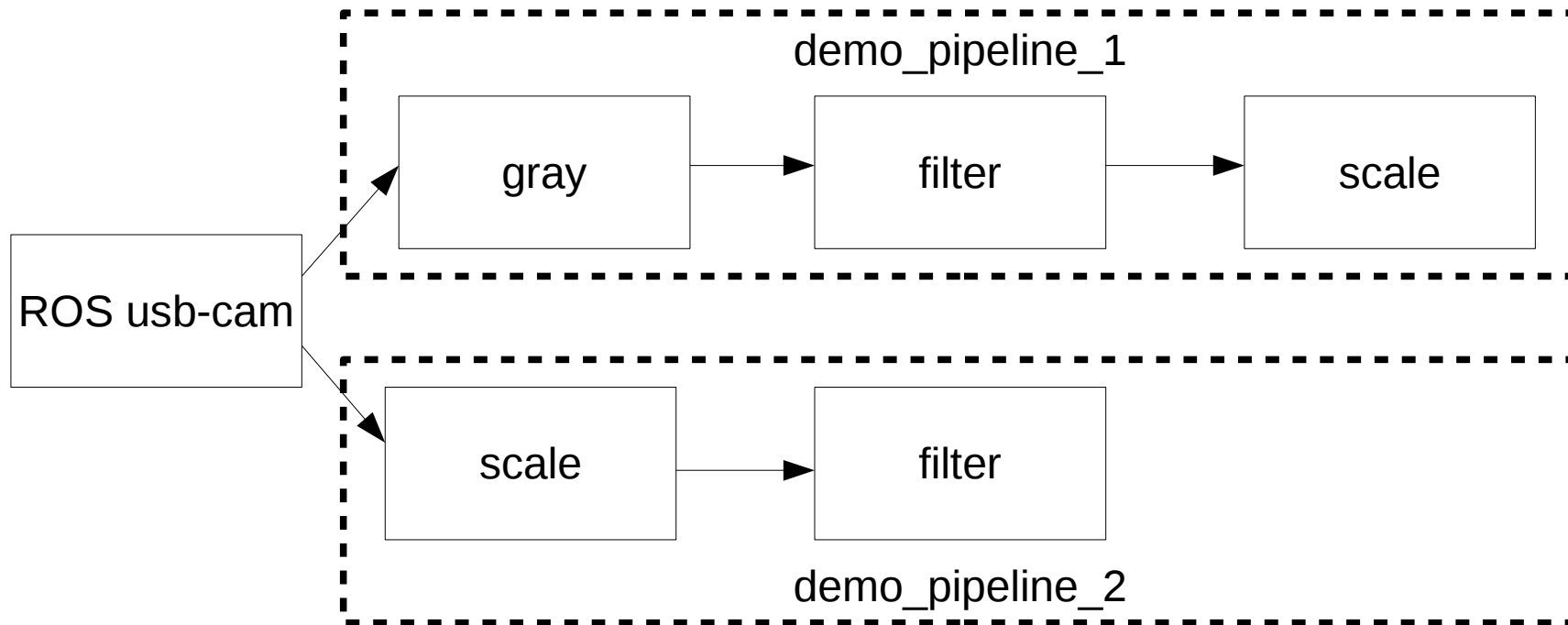
- 2 Nodelets (Plugins)

- Nachrichteninhalt kann kopiert oder Adresse darauf übergeben werden
- Abhängig von Publisher



- Nodelets sind Plugins mit der Basisklasse `nodelet::Nodelet` (siehe [Plugin-Workshop](#))
- Eigenschaften von Nodelets:
 - Durch die Plugin-Klasse:
 - Gemeinsamer Speicherbereich möglich
 - Kann dynamisch geladen und entladen werden
 - Durch die Nodelet-Basisklasse:
 - Emuliert Verhalten eines ROS Nodes
 - Kann mit `nodelet_manager` ohne eigenen Verwaltungs-Node gestartet werden

- Bsp.: https://github.com/martin-seidel/nodelet_cv_demo



- CMakeLists.txt
 - find_package(... roscpp nodelet)
 - add_library(**NodeletLibName** src/**nodelet_src_name**.cpp)
 - target_link_libraries(**NodeletLibName** \${catkin_LIBRARIES})
- package.xml
 - <build_depend>nodelet</build_depend>
 - <run_depend>nodelet</run_depend>
 - <export>
 - <nodelet plugin="\${prefix}/nodelet_plugins.xml" /></export>

```
1 <library path="lib/libNodeletLibName">
2   <class
   name="nodelet_namespace/NodeletClassName"
   type="nodelet_namespace::NodeletClassName"
   base_class_type="nodelet::Nodelet">
3     <description>
4       The function of the nodelet.
5     </description>
6   </class>
7 </library>
```

```
#pragma once
```

```
#include <nodelet/nodelet.h>
```

```
...
```

```
namespace nodelet_namespace {
```

```
    class NodeletClassName : nodelet::Nodelet {
```

```
    public:
```

```
        NodeletClassName();
```

```
        virtual ~NodeletClassName();
```

```
        virtual void onInit();
```

```
    };
```

```
}
```

```
#include ".h"

#include <pluginlib/class_list_macros.h>

PLUGINLIB_EXPORT_CLASS(nodelet_namespace::NodeletCl  
assName, nodelet::Nodelet)

namespace nodelet_namespace {

    NodeletClassName::NodeletClassName() {}

    NodeletClassName::~~NodeletClassName() {}

    void NodeletClassName::onInit() {
        //Initialisierung, die normalerweise im Konstruktor steht
    }

    ...
}
```


- Welche Plugins sind für die nodelet-Basisklasse registriert:
 - *rospack plugins --attrib=plugin nodelet*
- Starten im Launchfile:
 - `<node pkg="nodelet" type="nodelet" name="nodelet_manager_name" args="manager" output="screen"/>`
 - `<node pkg="nodelet" type="nodelet" name="NodeletRuntimeName" args="load nodelet_namespace/NodeletClassName nodelet_manager_name" output="screen">
 <!-- Parameter im Nodelet-Namespace des Parameterservers -->
 <param name="param1" value="text"/>
 <param name="param2" value="3.8"/>
</node>`

- Aus eigenem Namespace des Parameterserver

```
std::string string_param;  
getPrivateNodeHandle().getParam("param1", string_param);
```

- Überprüfen ob ein Parameter global gesetzt wurde

```
if (getPrivateNodeHandle().hasParam("/param1")) {...}
```

- Parameter mit Standardwert einlesen

```
float float_param;  
getPrivateNodeHandle().param<float>("param2",  
float_param, "1.0");
```

- Weitere Infos in der [ROS-Wiki](#)
-