

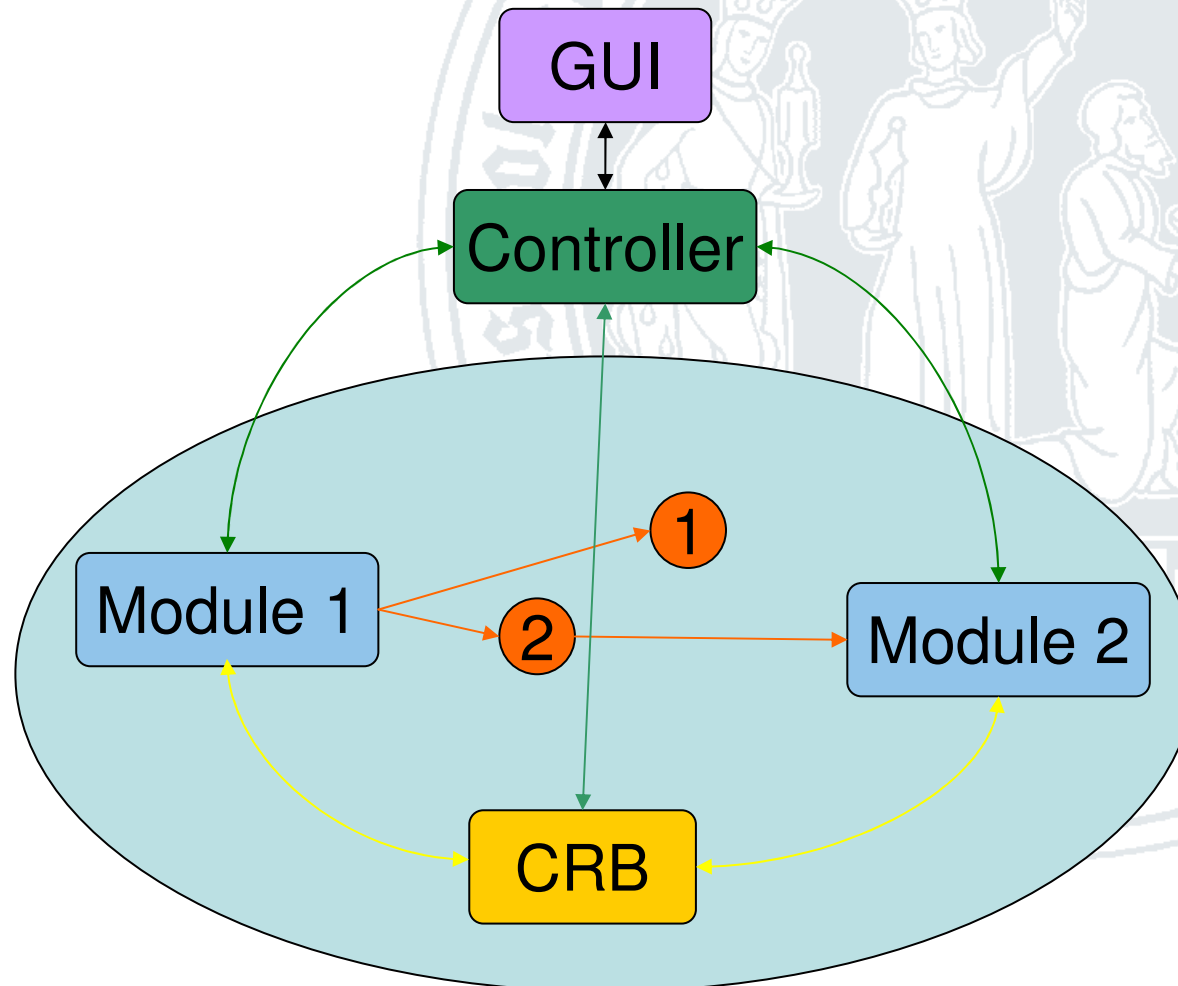
# Kollaboratives Arbeiten auf dem Desktop

- **Distributed Working**
- **Collaborative Working**
- **Mirroring (Spiegeln von Pipelines)**
- **Scripting Interface**

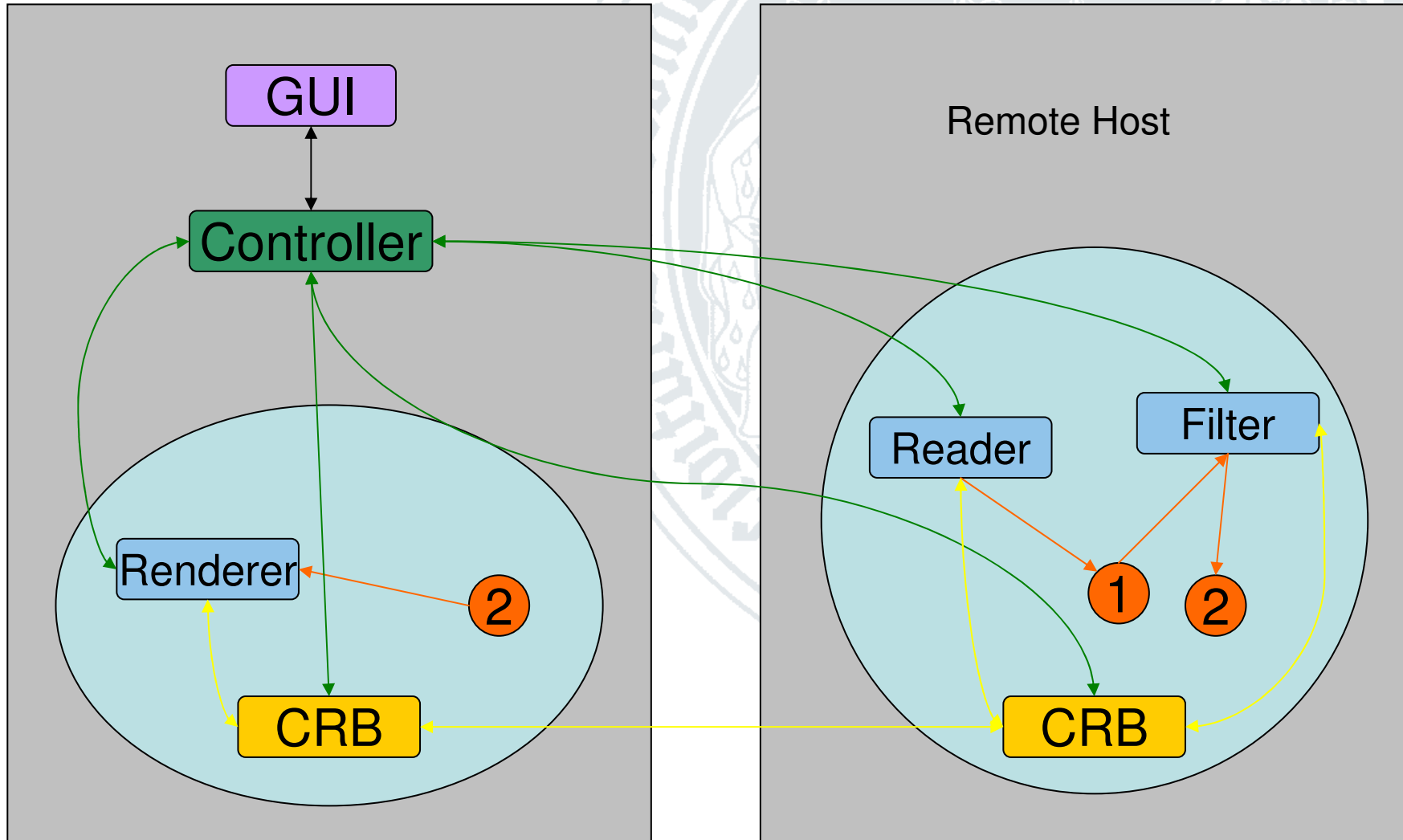
# Distributed Working

1. **Hinzufügen von Modulen, die auf anderen Rechnern laufen**
2. **Vorteile:**
  - **Benutzen von Modulen, die lokal nicht vorhanden sind**
  - **Bessere Ausnutzung der CPU Ressourcen**
    - Nutzung leistungsfähigerer Hardware
    - Nutzung mehrerer Rechner gleichzeitig
  - **VR-Renderer auf dediziertem Grafikrechner**

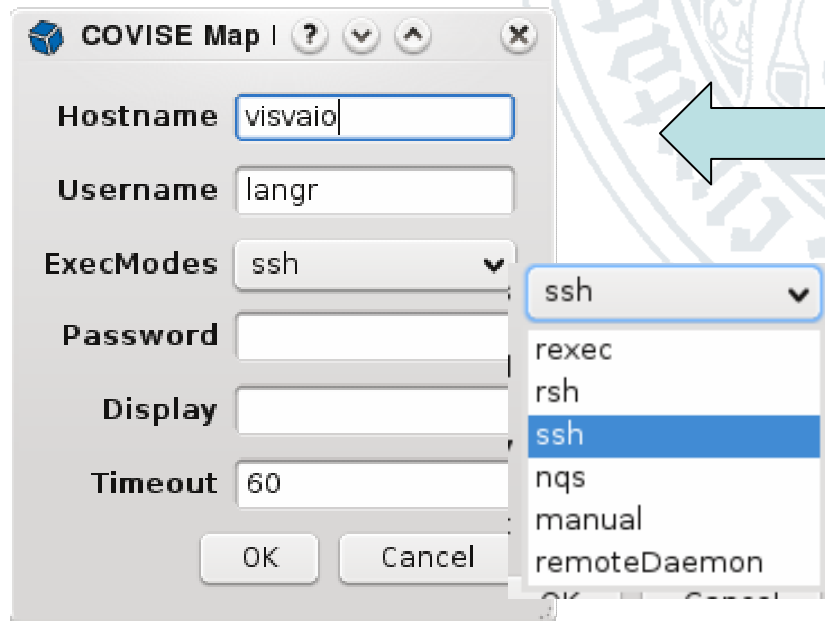
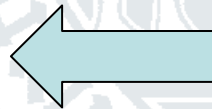
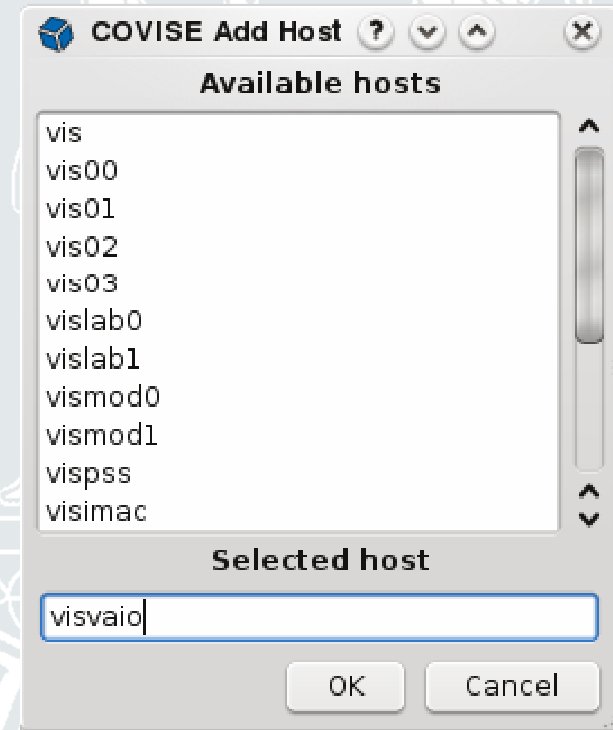
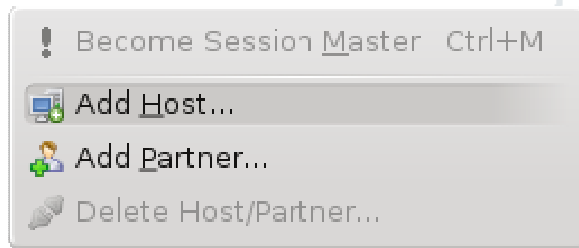
# Lokale Sitzung



# Distributed Working



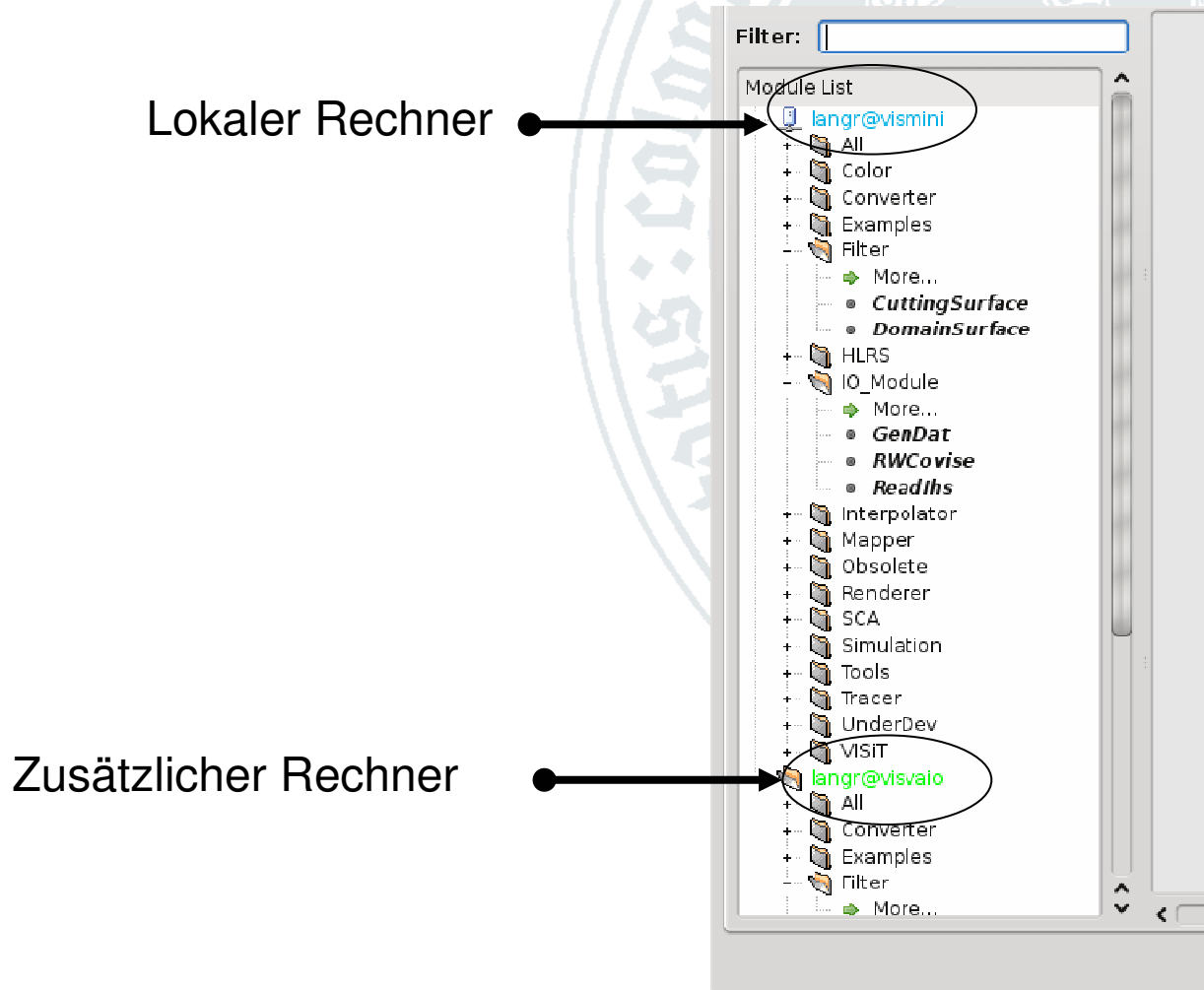
# Distributed Working: Add Host



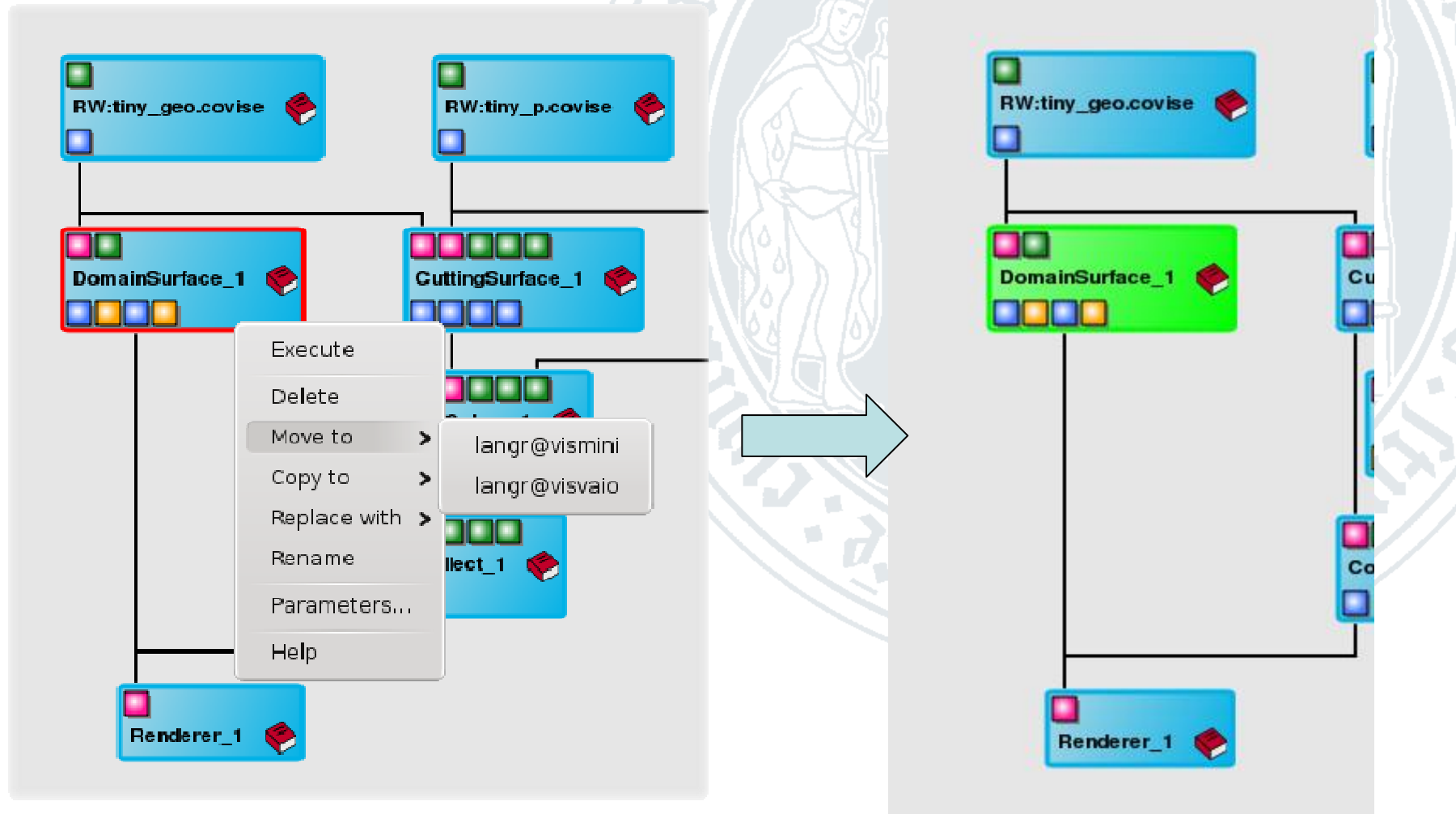
# Execution Modes

- **rexec (veraltet)**
  - Benötigt Benutzername und Passwort, ähnlich wie telnet
- **rsh**
  - benötigt nur Passwort, rsh Regeln einhalten
- **ssh**
  - benötigt nur Passwort und ssh
- **nqs**
  - NEC Queing-System
- **RemoteDaemon**
  - Setzt einen gestarteten „RemoteDaemon“ auf der Gegenseite voraus
- **manual**
  - Der CRB auf dem zusätzlichen Rechner muss von Hand gestartet werden. Die einzugebende Zeile wird in der „Message Area“ ausgegeben.  
*start „crb 31005 134.95.115.197 1005“*
  - Sinnvoll bei Firewallproblemen oder nicht vorhandenem Account auf der Gegenseite

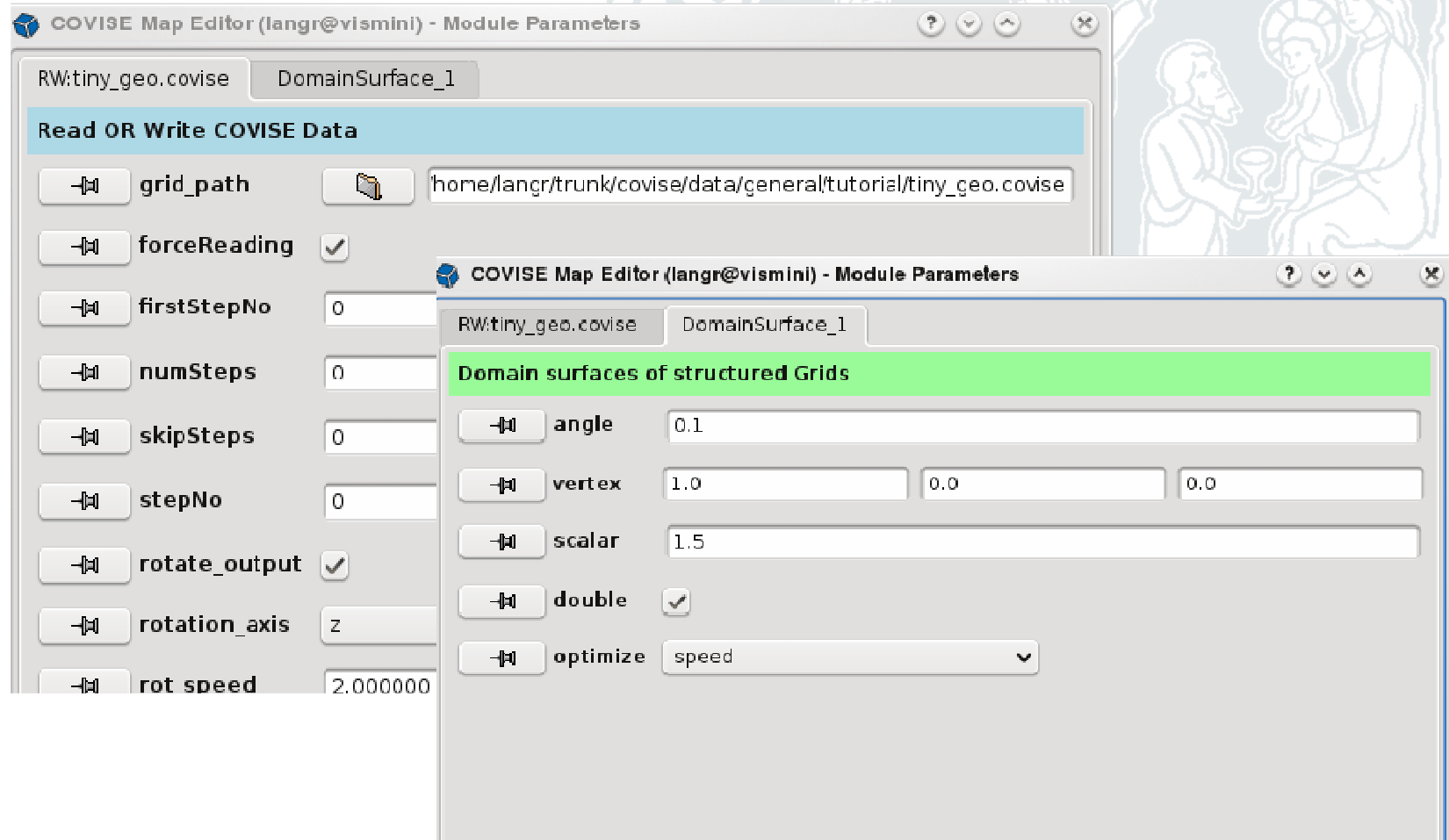
# Distributed Working: Add Host



# Verschieben eines Modules



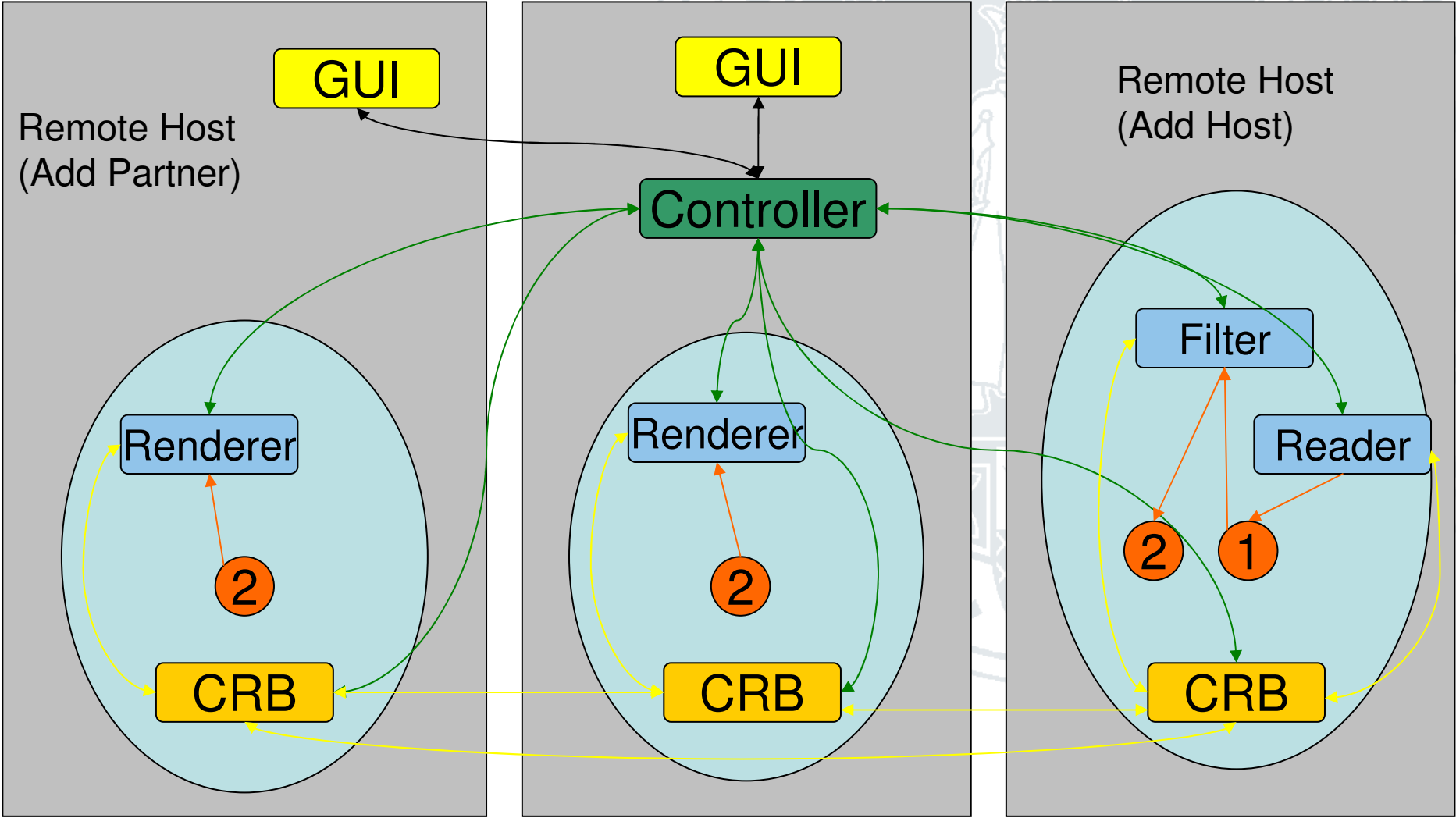
# Module Parameter Fenster



# Collaborative Working

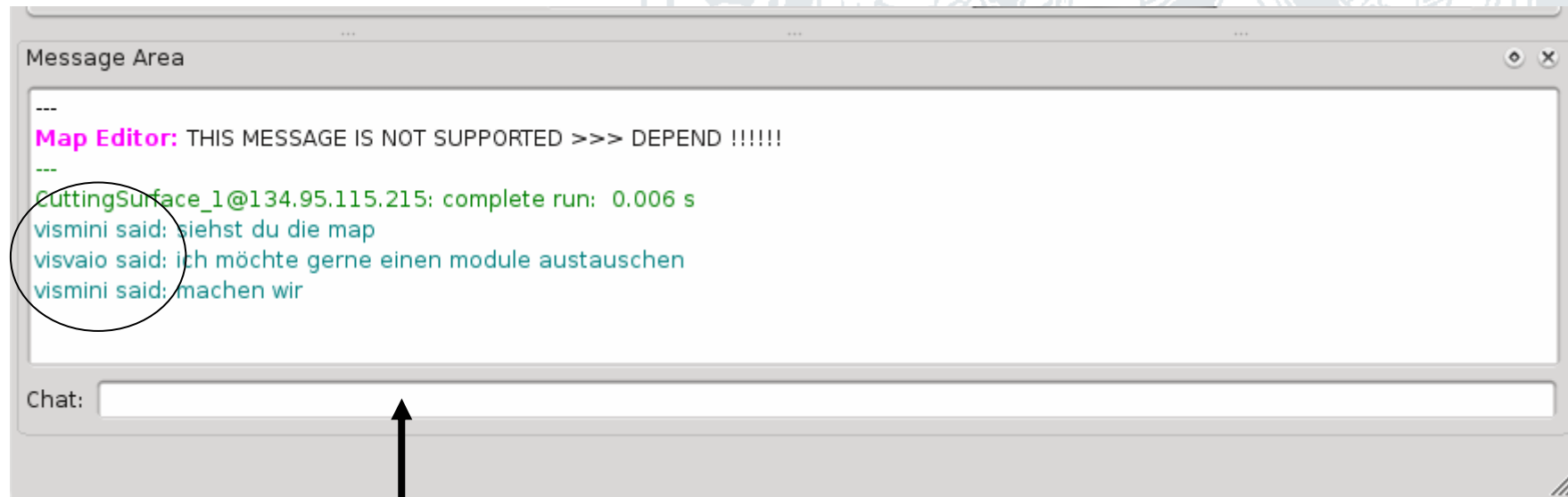
1. Hinzufügen von Modulen, die auf anderen Rechnern laufen
2. Für jeden Partner gibt es ein Userinterface
3. Bei jedem Partner wird automatisch eine Instanz für jeden Renderer gestartet
4. Parameter werden sofort automatisch aktualisiert
5. Alle Partner betrachten dasselbe Netzwerk
6. Vorteile :
  - Wie bei „Distributed Working“
  - Gemeinschaftliches Arbeiten an Netzwerken und Daten
  - Kommunikation durch „Chat“ Möglichkeit

# Collaborative Working



# Collaborative Working

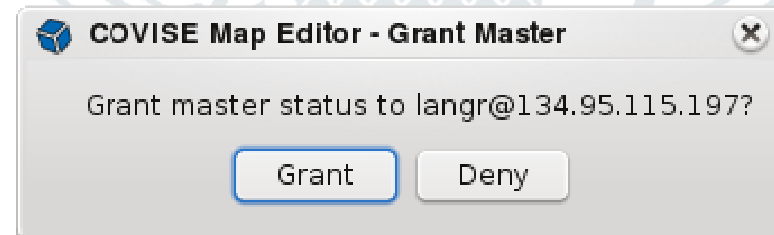
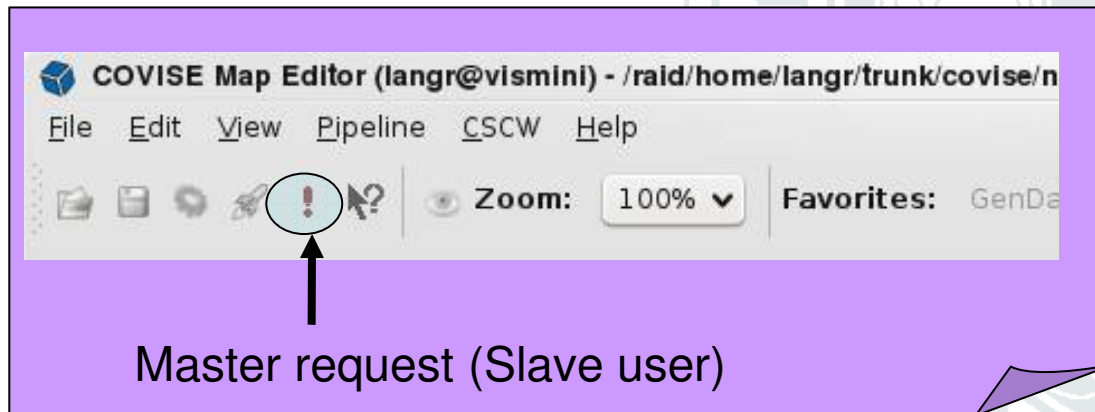
Die Message Area wird standardmäßig eingeblendet



Chatline

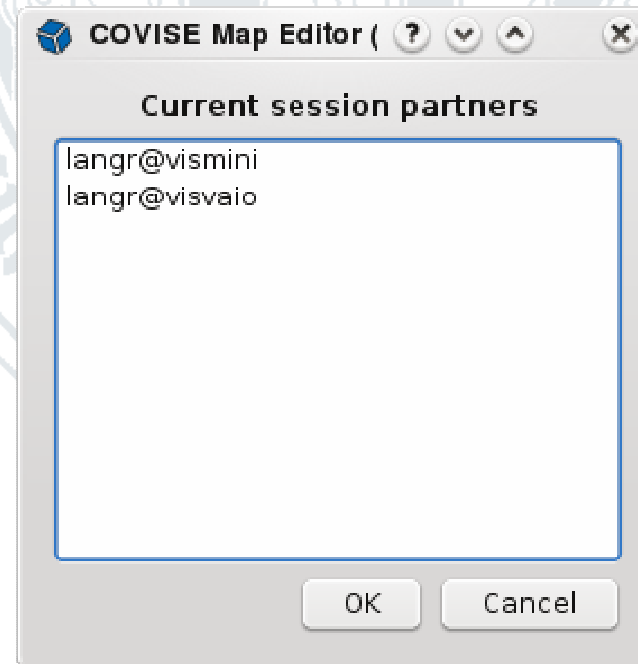
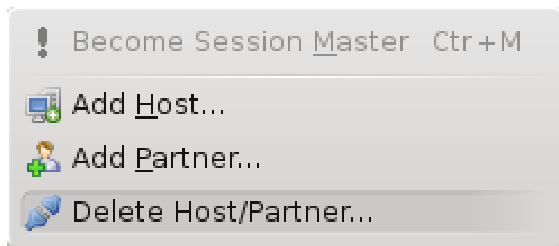
# Collaborative Working

Wir werde ich Master ?



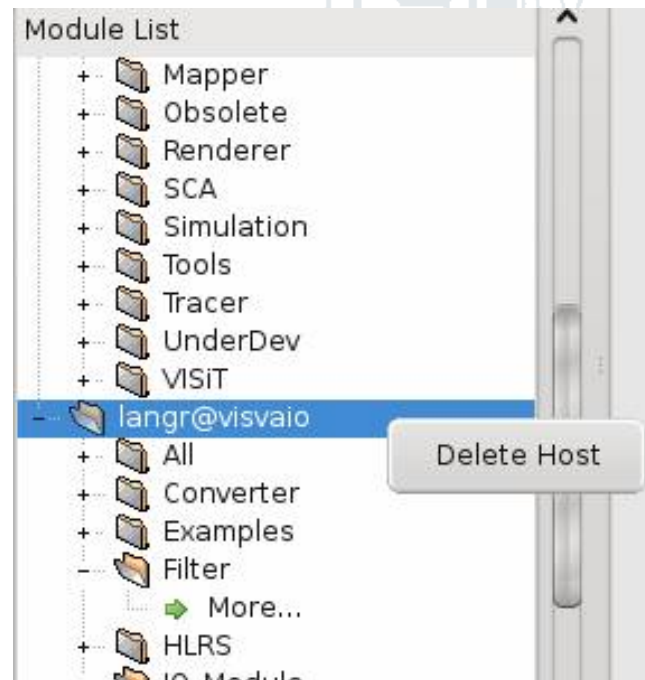
# Löschen eines Hosts

## 1. Methode: Über das CSCW-Menü



# Löschen eines Hosts

## 2. Methode: Rechtsklick auf Host im *Module Browser*



# Spiegeln von Netzwerken

- **Noch nicht komplett implementiert**
- **Netzwerke (Module) werden teilweise oder komplett auf andere Rechner gespiegelt**
- **Zweck :**
  - **Aufbrechen von Pipelines, um den Datenverkehr zu minimieren (Kommunikation <-> Neuberechnung)**

# Scripting Interface

- **Basierend auf Python**
- **Start durch `covise [-script] fname.py`**
- **Erzeugen von Pythonfiles durch**
  1. `map_converter -P -o converted.py`  
`$COVISEDIR/net/tutorial/channel.net`
  2. **Abspeichern einer Map als `fname.py` statt `fname.net`**

# Vergleich .py und .net maps

```
• # create global net
• #
• network = net()
• #
• # MODULE: GenDat
• #
• GenDat_1 = GenDat()
• network.add( GenDat_1 )
• GenDat_1.setPos( 22, 22 )
• #
• # set parameter values
• #
• GenDat_1.set_Coord_Type( 1 )
• GenDat_1.set_Coord_Representation( 1 )
• GenDat_1.set_Coord_Range( 1 )
• GenDat_1.set_Function( 1 )
• .
• .
• #
• # MODULE: Collect
• #
• Collect_1 = Collect()
• network.add( Collect_1 )
• Collect_1.setPos( 22, 220 ) .
• .
• .
• #
• # CONNECTIONS
• #
• network.connect( GenDat_1, "GridOut0", DomainSurface_1, "GridIn0" )
• network.connect( GenDat_1, "DataOut0", Colors_1, "DataIn0" )
• #
• .
• .
```

```
• #630
• 1
• LOCAL
• LUSER
• 5
• GenDat
• 1
• LOCAL
• IO_Module
• TITLE=GenDat_1
• 22
• 22
• 0
• 3
• GridOut0
• UniformGrid|RectilinearGrid|StructuredGrid
• Grid
• Default
• GenDat_1_OUT_00
• DataOut0
• Float
• Scalar Data
• Default
• GenDat_1_OUT_01
• DataOut1
• Vec3
• Vector Data
• Default
• GenDat_1_OUT_02
• 12
• Coord_Type
• Choice
• Coordinate type
• 1 Uniform Rectilinear Random Half_cyl Full_cyl Torus
• IMM
• -1
```

# Pythonbefehle

- `myNet = net()` Initialisieren einer Map
- `gendat=GenDat()` Starten eines Modules
- `myNet.add(gendat)` Hinzufügen des Moduls zur Map
- `gendat.showParams()` Anzeigen der Parameter
- `gendat.showPorts()` Anzeigen der Ports
- `gendat.set_Function(3)` Setzen eines neuen Wertes
- `renderer=QtRenderer()`
- `myNet.add(renderer)`
- `myNet.connect(gendat, "GridOut0", renderer, "RenderDat")`
- `runMap()` Ausführen der Map