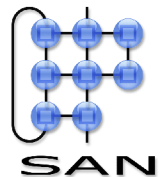


# Health Management in ....



# Application domain

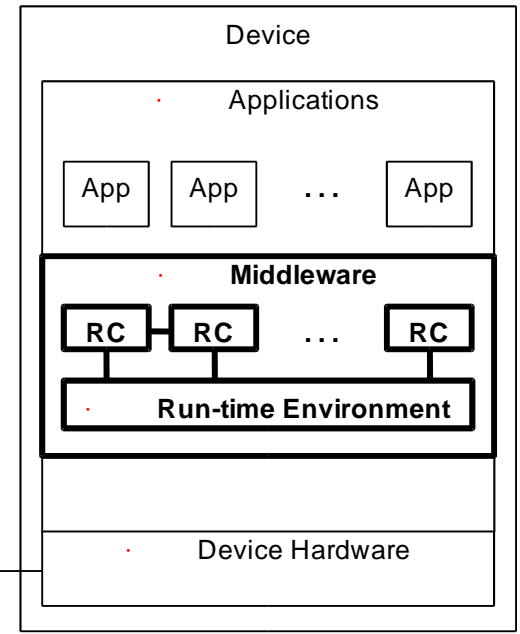
## High Volume Consumer Electronics



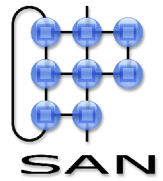
# Scope: Component Based Middleware



TU/e



The middleware must enable:  
**robust** and **reliable** operation  
**upgrading** and **extension**  
component trading



# Robocop Component



## Robocop Component

Resource Model

Simulation Model

Documentation

Functional Model

Source Code

...

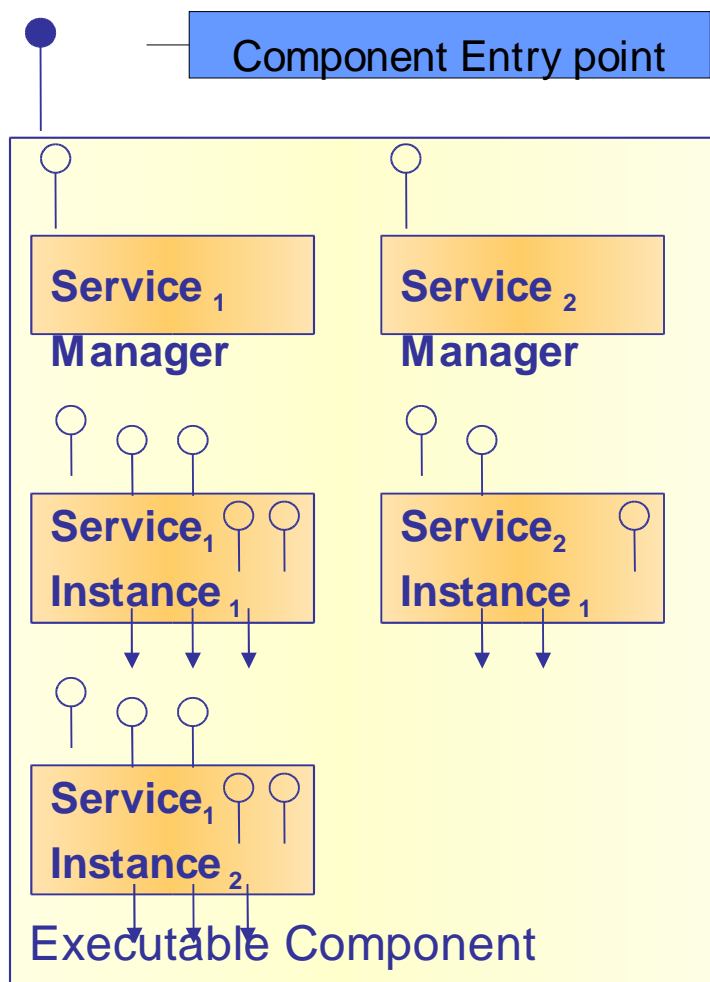
Executable Component

A Robocop Component is a set of Models

- Model can be machine and/or human readable / executable
- Typically: one of those is executable on a target system  
Called the (executable) component
- No definition of the packaging yet  
(such as .NET assembly)  
(thinking about e.g. XML)



# Robocop Executable Component

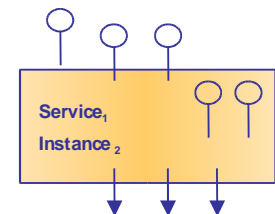
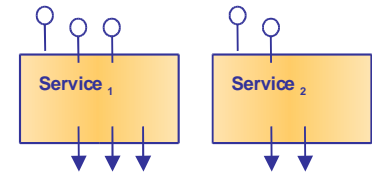
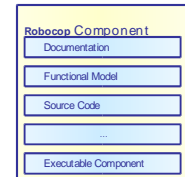


- Services have explicit dependencies
  - Requires interfaces
- Services have interface for managing dependencies
- Services are instantiated at Run Time
- Service Manager is responsible for
  - Instantiating services
  - Presetting attributes
- Service Instance is an entity with its own data (state) and a unique identity
- Components are instantiated in OS terms
  - Static in process (LIB)
  - Dynamic in process (DLL)
  - Dynamic out process (EXE)
- Component Entry point provides fixed entry points to
  - Register to Run Time
  - Service Manager

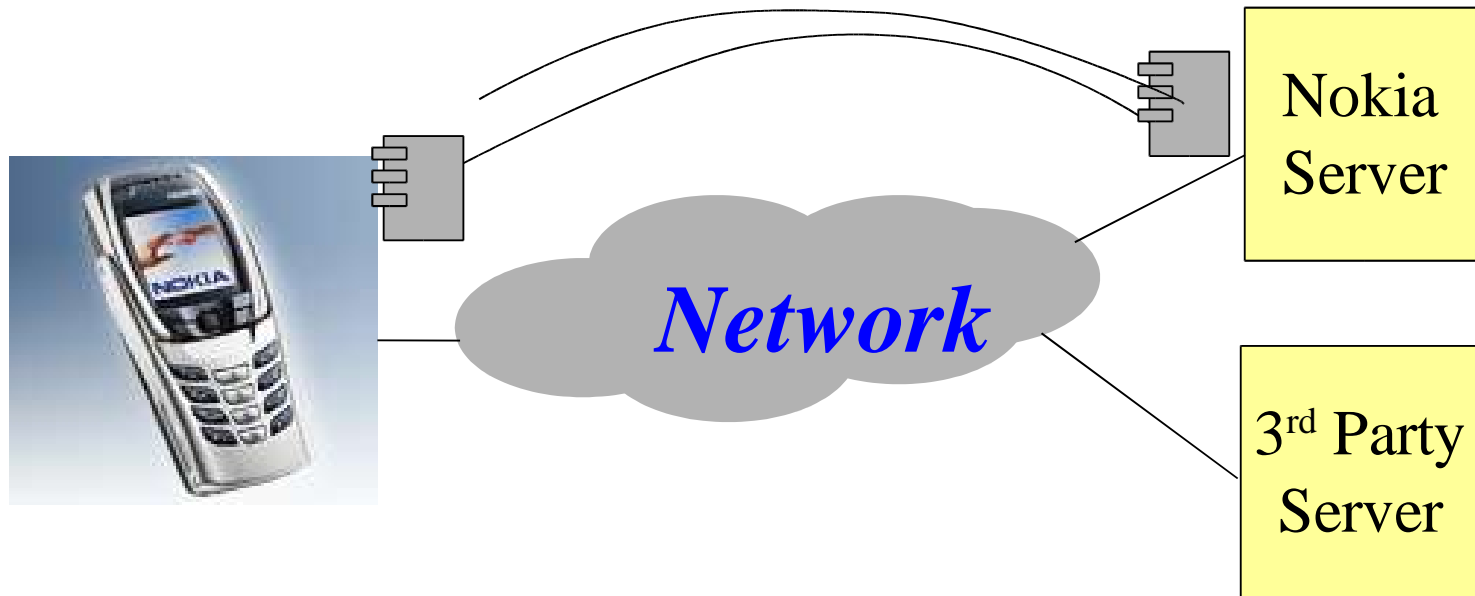
# Things to remember about Robocop



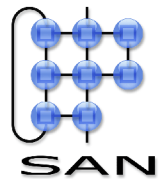
- A Robocop Component is a set of models
- Execution component contains a set of Services
- Service has set of Interfaces (provides & requires)
- An Interface has a set of Operations
- At runtime Services are instantiated  
(Service Instance  $\approx$  object)



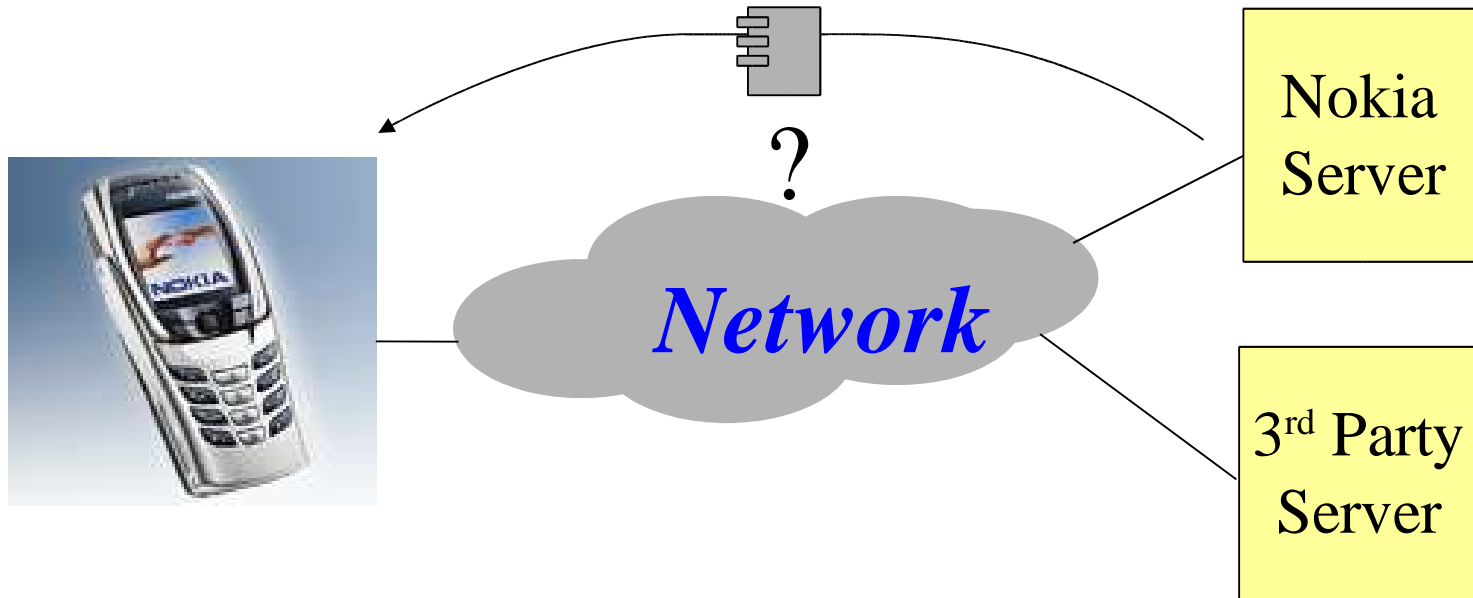
# Goal: Maintain System Integrity



During component addition,  
removal,  
replacement

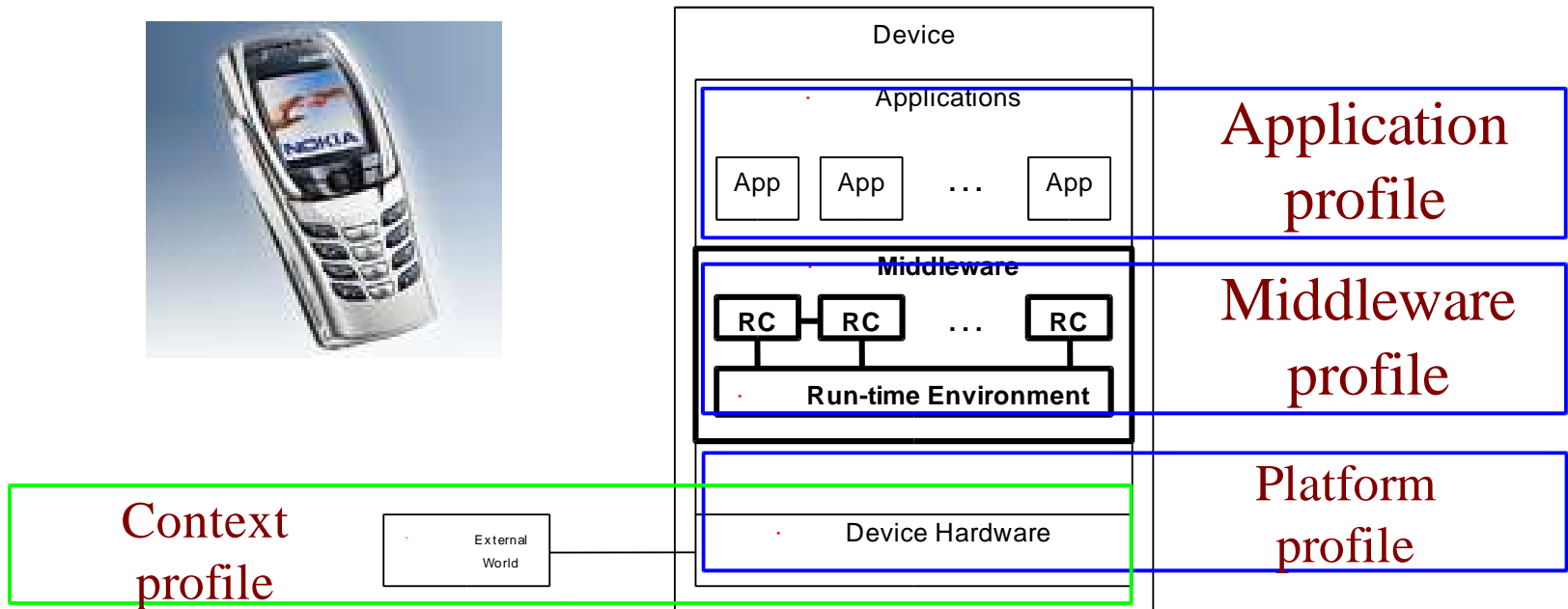


# Questions



- › Is current system configuration healthy?
- › Will system configuration be healthy after change X ?
- › What modifications to the system configuration are needed to cure a un-healthy system?

# Profiles for all !!

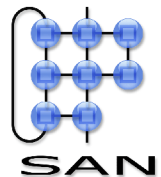
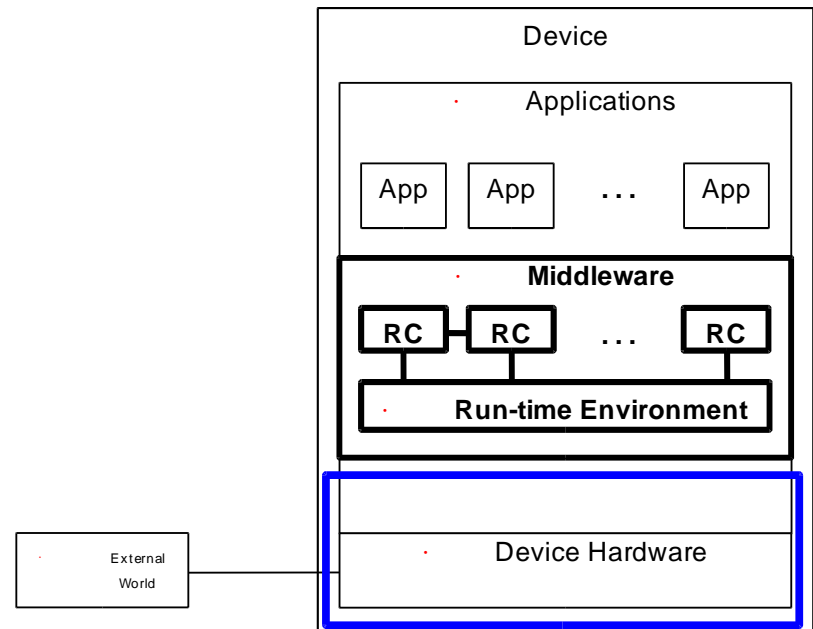


# Platform profile



Platform profile provides info on:

- Architecture
- CPU
  - Family
  - Model
  - ClockSpeed
  - Cache Size
- Storage
  - Memory
  - Swap
  - FileSystem
- OS
  - ID
  - Version

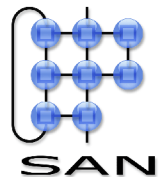


# Platform Example



## This notebook:

```
<device>
  <platform>
    <arch id='i386' />
    <cpu>
      <vendor id='GenuineIntel'>
      <family id='15' />
      <model id='2' name='Intel(R) Pentium(R) 4 Mobile CPU 2.00GHz' />
      <clockspeed mhz='1994.159' />
      <cache size='512KB' />
    </cpu>
    <storage>
      <memory size='261316608' />
      <swap size='534601728' />
      <filesystem name='/' size='9463MB' />
      <filesystem name='/boot' size='101MB' />
      <filesystem name='/local' size='8811MB'>
    </storage>
    <os name='GNU/Linux' version='2.4.20-18.9' />
  </platform>
</device>
```

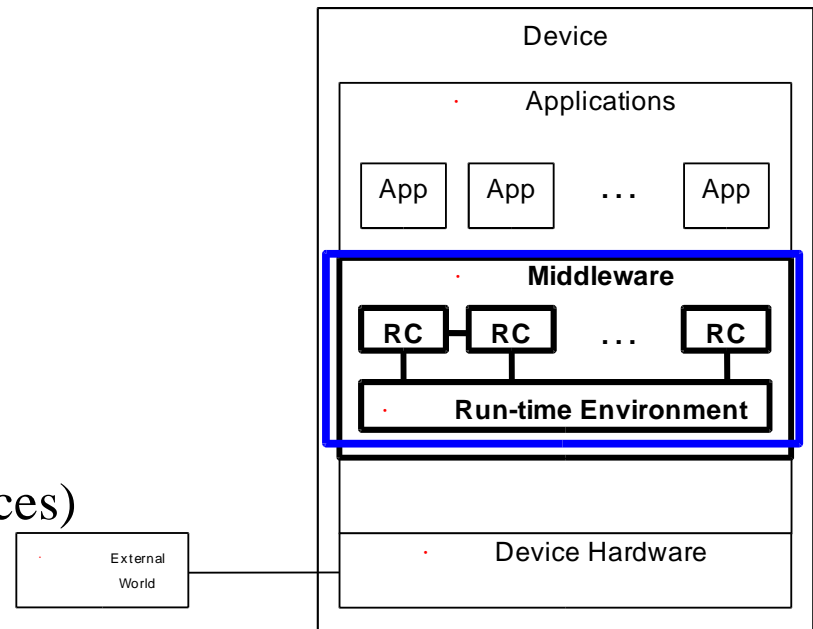


# Middleware Profile



Middle profile provides info on:

- Run time Environment
  - Version
  - QoS Support (Yes / No)
- Registered components
  - Location on the device
  - Services
- Complies relations (between services)



# Middleware Example

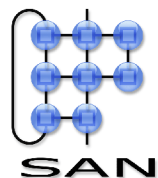


## Part of my middleware:

```

<middleware>
  <rre version='0.42a' />
  <component guid='BCF018E0-01CF-D711-87C6-0008744C31AC'>
    <location url='file:/libadvancedcomputerplayers.so' />
    <service guid='20688A2F-01CF-D711-87C6-0008744C31AC' />
  </component>
  <component guid='B7621504-7FCD-42EA-BCF0-90F67FE557C7'>
    <location url='file:/libcomputerplayers.so' />
    <service guid='6782A98F-06D5-436F-AE89-0D6E064AB047' />
  </component>
  ..
  <complies>
    <complies_relation from='20688A2F-01CF-D711-87C6-0008744C31AC' to='20688A2F-01CF-D711-87C6-0008744C31AC' />
    <complies_relation from='68CCA7C4-C24A-4BEE-9A5E-AF79B806483D' to='20688A2F-01CF-D711-87C6-0008744C31AC' />
    <complies_relation from='6782A98F-06D5-436F-AE89-0D6E064AB047' to='6782A98F-06D5-436F-AE89-0D6E064AB047' />
  ..
  </complies>
</middleware>

```

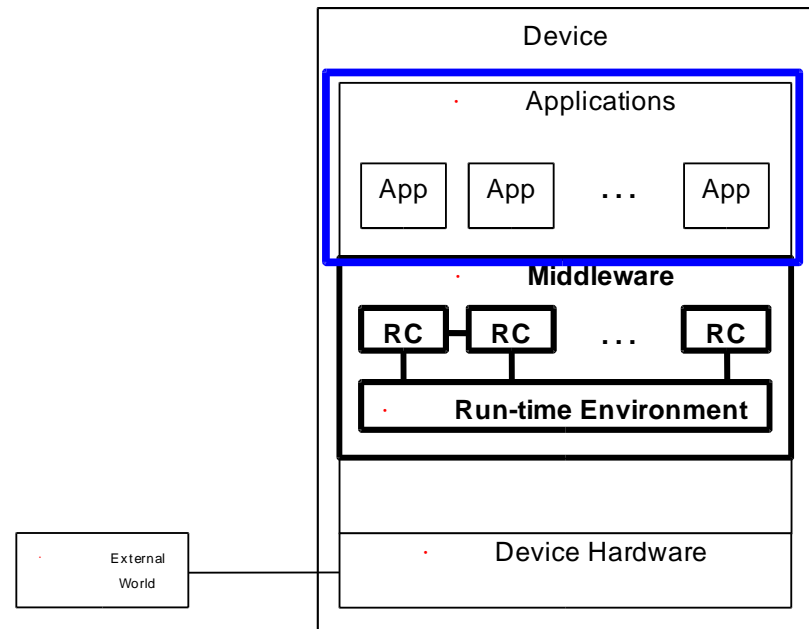


# Application profile



Application profile provides info on:

- Available applications
  - Location on device
  - Dependencies on services
- Run Time Structure
  - Service Instances
  - How they are bound



# Application Example

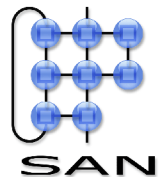


## My Robocop application (the only one):

```

<application name='TicTacToe' version='1.0'>
  <location url='file:/gametestapp'>
    <dependencies>
      <service guid='AB98B03D-8F5A-4EC7-9064-7010D0601C75' />
      <service guid='3341EC0A-175A-47D3-8BA2-367999D71B71' />
      <service guid='A91C7422-DF1D-451B-9052-AE511A23CB8F' />
      <service guid='68CCA7C4-C24A-4bee-9A5E-AF79B806483D' />
    </dependencies>
    <structure>
      <instances>
        <service_instance guid='AB98B03D-8F5A-4EC7-9064-7010D0601C75' ref='display' />
        <service_instance guid='68CCA7C4-C24A-4bee-9A5E-AF79B806483D' ref='manager' />
        ...
      </instances>
      <binding>
        <bind provides_ref='display' requires_ref='manager'
              provides_port='game_output' requires_port='output' />
        ...
      </binding>
    </structure>
  </application>
  ..

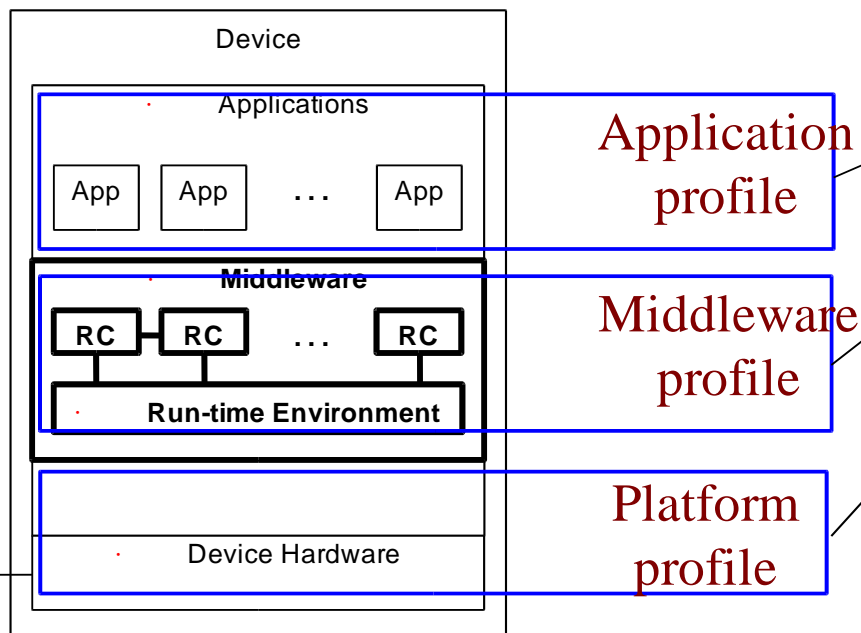
```



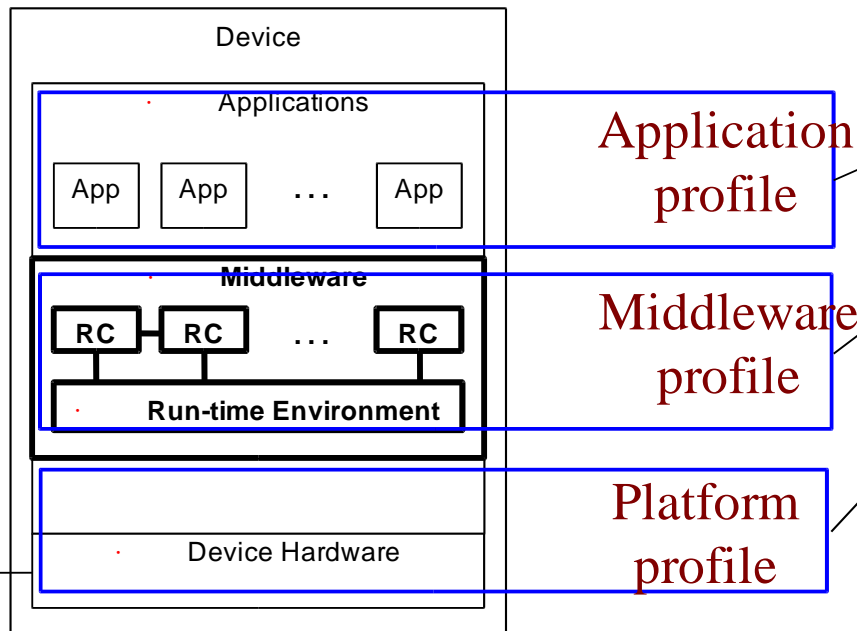
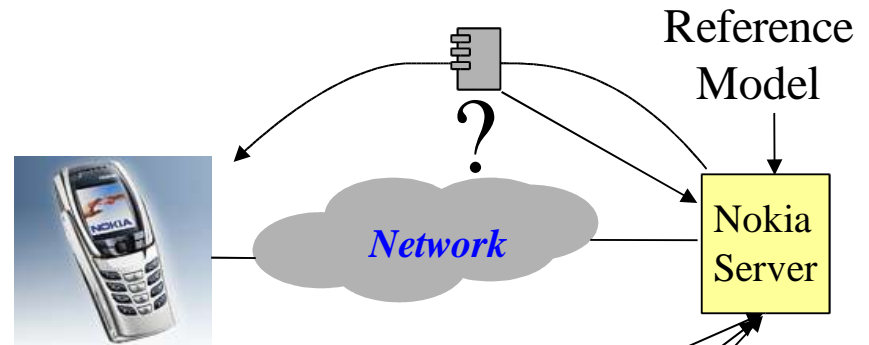
# Is this device healthy?



Reference  
Model



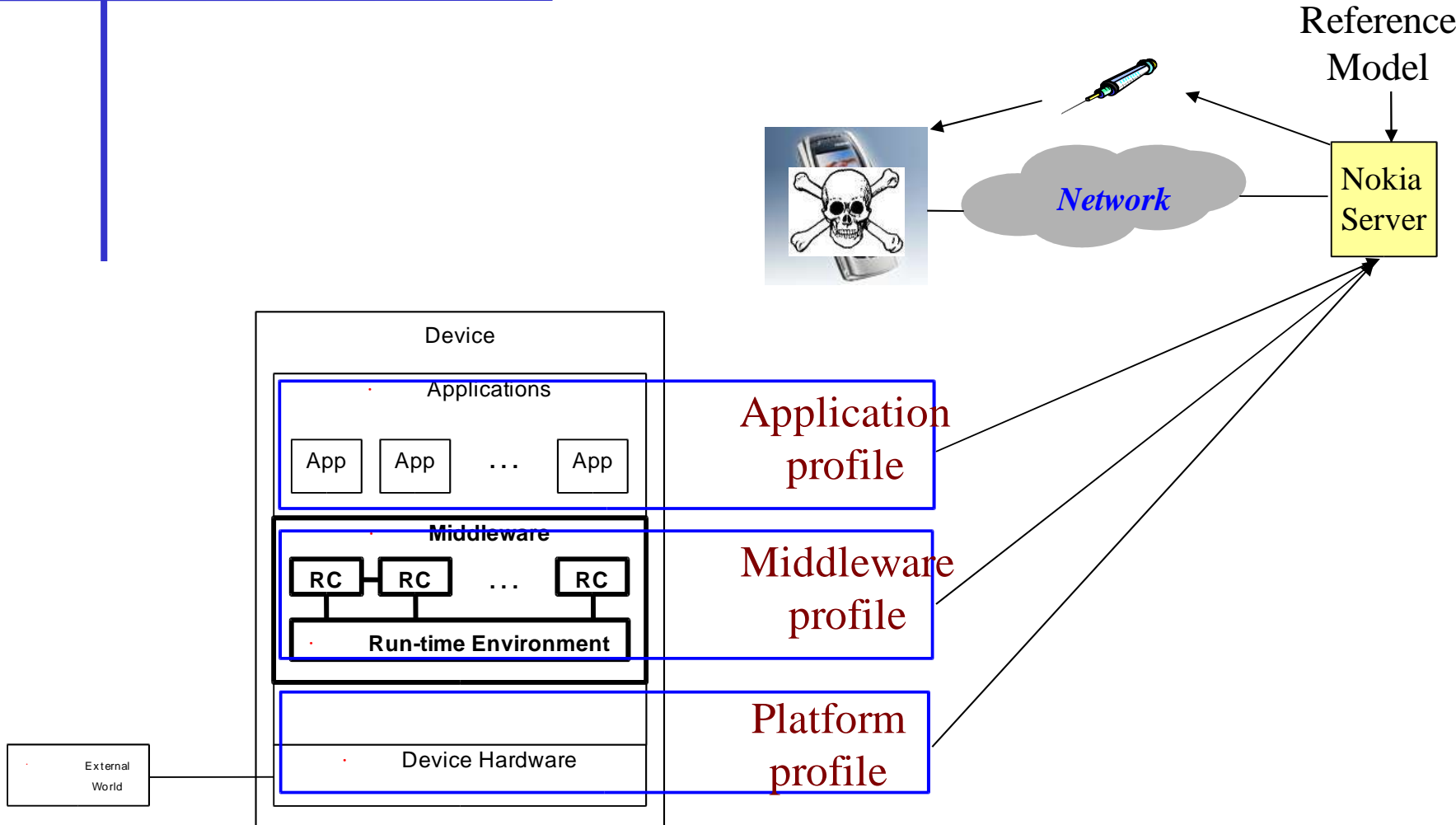
# Will device be healthy after change X ?



# What modifications are needed to cure device ?



TU/e





# May the force be with Space4U

If you have any ideas please contact [j.muskens@tue.nl](mailto:j.muskens@tue.nl). Only ideas that are fully implemented and validated will be accepted :)

