

# **Einstieg in CFEngine**

## Intro

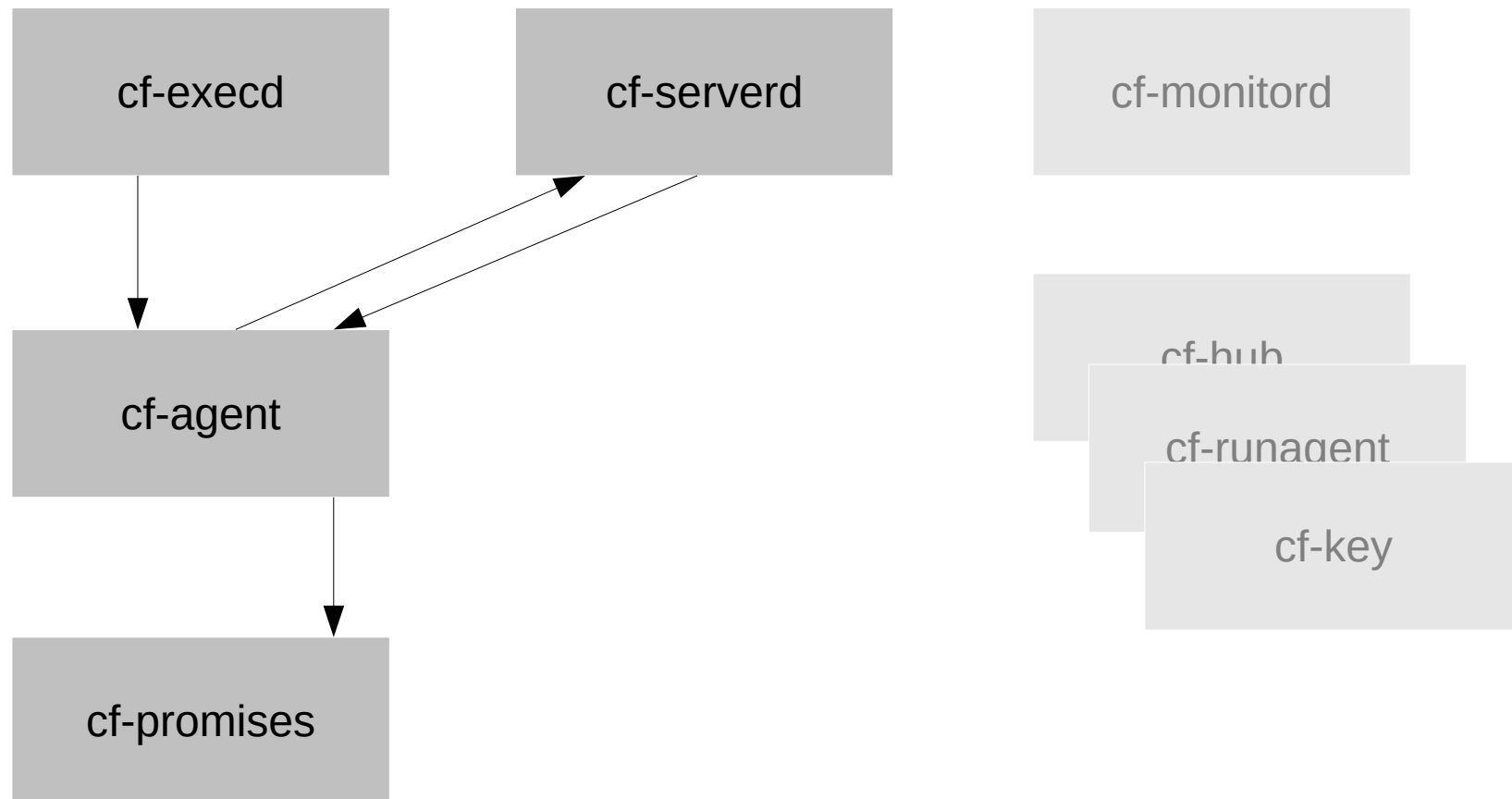
- Do:
- Funktionskonzept
  - Begriffe
  - Beispiele Beispiele Beispiele (WIE)
- Don't:
- Vergleich mit anderen ConfigTools
  - DesignCenter und kommerzielle Version
  - Inhalte (WAS)

## CFEngine, Konzept, Begriffe, Übersicht

- promise theory
- Agenten die freiwillig kooperieren
- desired state configuration, deklarativ nicht imperativ
- convergent iterations
- cf-execd, cf-serverd, cf-monitord
- cf-promises, cf-agent

# CFEngine, Konzept, Begriffe, Übersicht

## - components



## CFEngine, Konzept, Begriffe, Übersicht

- promises: ALLES ist ein "promise"
- "bundles" gruppieren promises
- "bodies" spezifizieren Attribute von promises
- promise types: vars, files, services, commads, classes....
- normal ordering: standard Reihenfolge der promise types
- class based descision making
- implicite looping: automatische Iteration über Listen

# CFEngine, Konzept, Begriffe, Übersicht

- syntax

```
bundle <type> <name> (<parameter>)  
{  
    <promise_type>:  
        <class>::  
            "<promisser>"  
                <attribute_1>    => "<value>",  
                <attribute_2>    => <body_name(<parameter>)>;  
}
```

```
body <type> <name> (<parameter>)  
{  
    <attribute_1>    => "<parameter>";  
}
```

## CFEngine, Konzept, Begriffe, Übersicht

- normal ordering: standard Reihenfolge der promise types  
INNERHALB eines "bundles" (Auszug)

vars

classes

files

packages

methods

processes

services

commands

reports

## CFEngine, Konzept, Begriffe, Übersicht

- class based descisions
- hard classes (discovered classes)
  - `cf-promises -v | egrep "hard class"`
- soft classes (defined in promisses)
  - `cf-promises -v | egrep "soft class"`
- `cf-promises --show-classes / --show-vars`



# CFEngine, Konzept, Begriffe, Übersicht

- implicite looping
  - data types: scalar (single value)  
list (collection of scalars),  
and data (json doc)
  - scalar types: string, int, real
  - ist ein promisser eine Liste, wird das promise für jedes Element der Liste evaluiert