

Test-Driven DevOps

mei^{ss}sa GmbH

Michael Jerger



Agenda

- Vorstellung
- Tests für Developer

Herausforderung Operations

- Ansatzpunkte für agile DevOps
- Testen mit dda-pallet



Agenda

- **Vorstellung**
- Tests für Developer

Herausforderung Operations

- Ansatzpunkte für agile DevOps
- Testen mit dda-pallet



meissa GmbH?

- ♦ Fokus: DevOps, Maintainer von dda-pallet
- ♦ Sitz Reutlingen, SW-Entwicklung Java & Uni Tübingen
- ♦ Arbeiten dezentral, d.h. 80% Remote-Arbeit
- ♦ Zusammenarbeit auf Augenhöhe

<https://www.meissa-gmbh.de>



Was ist dda-pallet?

- ◆ Funktionales DevOps System
- ◆ Test Driven
- ◆ Schnörkellos und direkt
 - ◆ Linux-server: functional - bash.
 - ◆ Cloud services: functional - api.
- ◆ Adressiert den kompletten LifeCycle, OpenSource, cloud zentriert, ...

meiSSa GmbH



... pallet ...

... ist ein clojure
DevOps Framework
von Hugo Duncan.

meiSSa GmbH

... wir komponieren pallet zu einem
DevOps System.



... dda ...

- ◆ DomainDrivenArchitecture
DDD Prinzipien in der funktionalen Welt:

- ◆ „Separate Domain from Infrastructure“

- ◆ „enable Domain Language“

Base & cloud Abstraktion = Infrastruktur

- ◆ Domain Language bestehend aus
 - ◆ Architektur Konventionen
 - ◆ Netzwerk-Layouts oder
 - ◆ Applikations Stacks



Agenda

- Vorstellung
- **Testing für Developer**

Herausforderung Operations

- Ansatzpunkte für agile DevOps
- Testen mit dda-pallet



Testarten für Dev

Test Technik / Test Umfang	Junit	Mockito	FitNesse	Selenium	Jmeter / LoadImpact	nmap / openVas
Unit						
Functional Integration						
Kunden-Interaktiv						
Infrastruktur Integration						

meiSSa GmbH



Agenda

- Vorstellung
- Tests in der Softwareentwicklung

Herausforderung Operations

- Ansatzpunkte für agile DevOps
- Testen in dda-pallet



naiv?

Naiiv landen wir komplett in
Infrastruktur Integration!

meiSSa GmbH



Ops = viel State

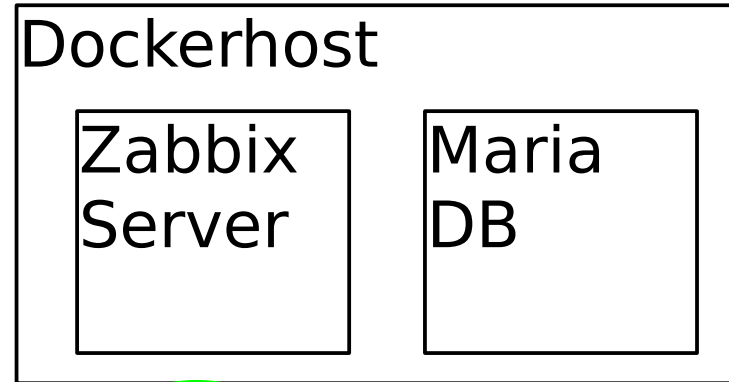
- ◆ Installierte Betriebssystem
- ◆ System-Konfiguration
- ◆ Update-Zustand
- ◆ Installierte Applikationen und deren Zustand
- ◆ Logfiles und deren Nutzung

meiSSa GmbH



State - was tun?

- State isolieren
 - Container



meiSSa GmbH

State explizieren

```
facts (let fn collect-netstat-fact [
  (collect-fact fact-id-netstat
    ("netstat" "-tulpen")
    :transform-fn parse-netstat))
```

Ops = Heterogene Technik

- Unix / Windows/ *sh / Powershell
- Perl / PHP / C / Java / Python / Ruby / GO / Lisp / JavaScript
- Make / ant / maven / gradle / lein / grunt
- Config von HW Firewall / Router / VPN
- Config von Applikationen

meiSSa GmbH



Heterogenität - was tun?

Heterogenität kapseln ,
(dünner) Anti Corruption Layer

Data-API

```
{:netstat-fact nil  
:file-fact (:root  
  "/etc"  
  "/absent")  
:netstat-fact (:suid {:"net" "2"})  
:file-test {:root-sth {:path "/root",  
  :exist? true},  
:etc {:path "/etc",  
  :exist? true},  
:absent {:path "/absent",  
  :exist? false}})
```

Bash-Interaktion

```
(defn collect-netstat-fact  
  "Defines the netstat resource.  
  This is automatically used to generate state data used  
  in tests."  
  (collect-fact :net-id-netstat ("netstat" "-tulpu") :transform-fn (fn [netstat]
```

meissa GmbH



Heterogenität - was tun?

Komplexität in übergreifendes Tool
verschieben

Domain:

```
(domain/create-stack-configuration
  [jem {vm-type : de}]
  [platform virtualbox
  vm-type : de]])
```

Infrastruktur:

```
{:meissa-managed-vm-group
  {:back name "managed-vm",
  :element [{:subdir-to-save ".ssh .mozilla .cupg"}]
  :name "user-home",
  :type :file-composed,
  :element [{:subdir-to-save ".ssh .mozilla .cupg"}],
  :generalized-on :source-system 1,
  :backup-user {:type "dataBackupSource",
  :encrypted-password "WIw...IUt2..."},
  :script-path "/usr/lib/dda/backup/"
  :meissa-managed-vm {:settings #(:install-vm-jdk
  :install-git
  :install-libreoffice
  :install-linux-basics
  :install-virtualbox-guest,
  :vm-user :jem,
  :desktop-wiki {:git-repo-urls ["repository.domaindrivenarchitecture
  :git-user-name "jem"],
  :meissa-settings #(:install-chromium
  :install-swappiness
  :install-redshift
  :install-telegram
  :install-gehaltstrenner},)
  :dda-user {:personal-key {:private-key "-----BEGIN RSA PRIVATE KEY-----\n.....\n---
  :public-key {:type "ssh-rsa",
  :comment "mje@jegerProject\n",
  :public-key "AAAAB3NzaC1yc2EAAAADAQABAAQCe0+
  :authorized-keys [{:type "ssh-rsa",
  :comment "mje@jegerProject\n",
  :public-key "AAAAB3NzaC1yc2EAAAADAQABAAQCe0+eiYDonq30
  :user-name "jem",
  :encrypted-password "rw4lq04w4e0FRd"}}
```

meissa GmbH

Ops = Infrastructure

- ◆ Netzwerk, Firewall, Routing, Cloud
- ◆ Integration in Ops Services
- ◆ Repositories: Linux-Packages, Software-Artefakte, Web-Downloads
- ◆ Security: Krypto-Keys & Credentials

meiSSa GmbH



Infrastructure - was tun?

Status	Pipeline	Commit	Pages	Time	Age
passed	#14993 by latest	master -> 5cb2a672		00:11:13	4 days ago
passed	#14933 by	master -> 330c...		00:12:30	4 days ago
failed	#1...7 by	master -> 92082...		00:02:19	4 days ago
passed	#1...5 by	master -> 4073...		00:11:38	5 days ago
passed	#14718 by	master -> e77c9fb0		00:00:33	1 week ago
passed	#14601 by	master -> e77c9fb0		00:00:34	1 week ago
passed	#13820 by	master -> ec968d36		00:09:46	2 weeks ago
passed	#13617 by	master -> eb643cd1		00:10:48	2 weeks ago

Infrastruktur Artefakte

continuous & getestete

meissa GmbH

Agenda

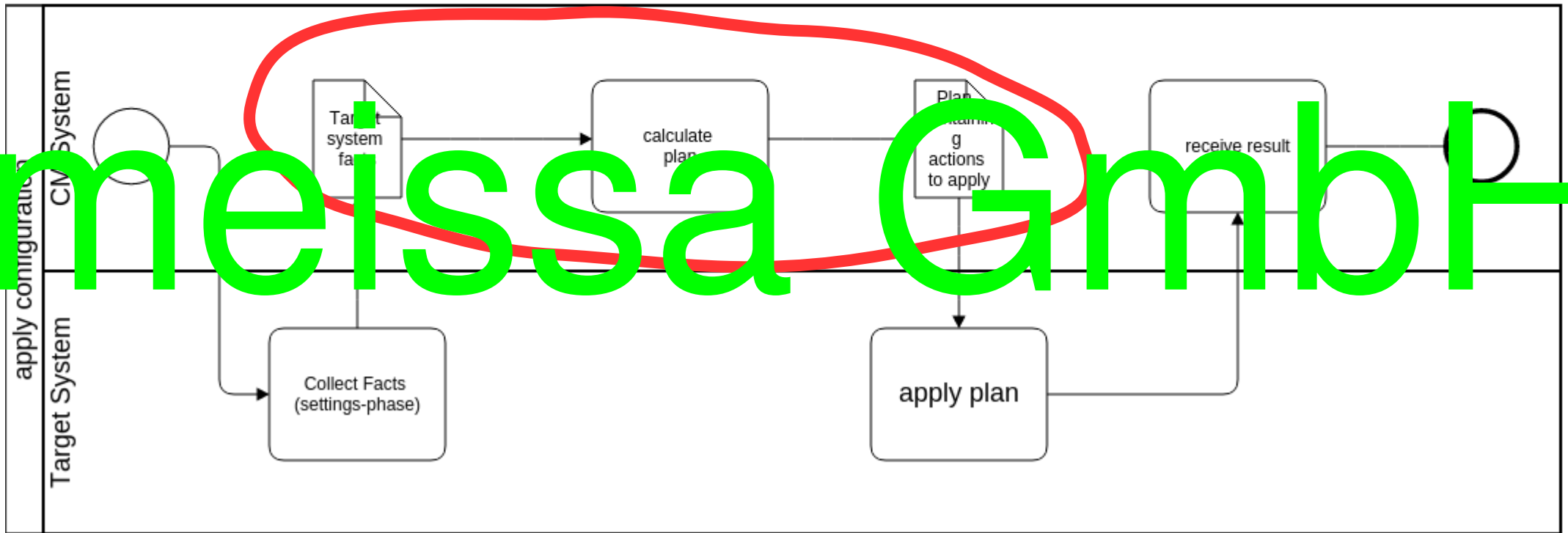
- Vorstellung
- Tests in der Softwareentwicklung

Herausforderung Operations

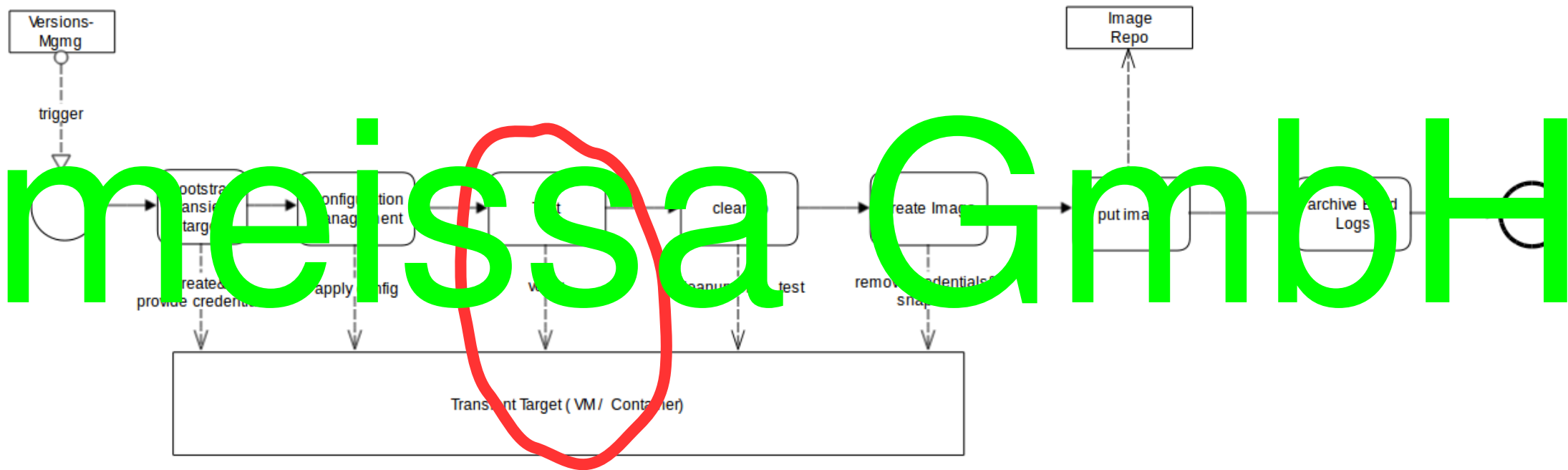
- **Ansatzpunkte für agile DevOps**
- Testen mit dda-pallet



Soll/Ist-Abgleich testen

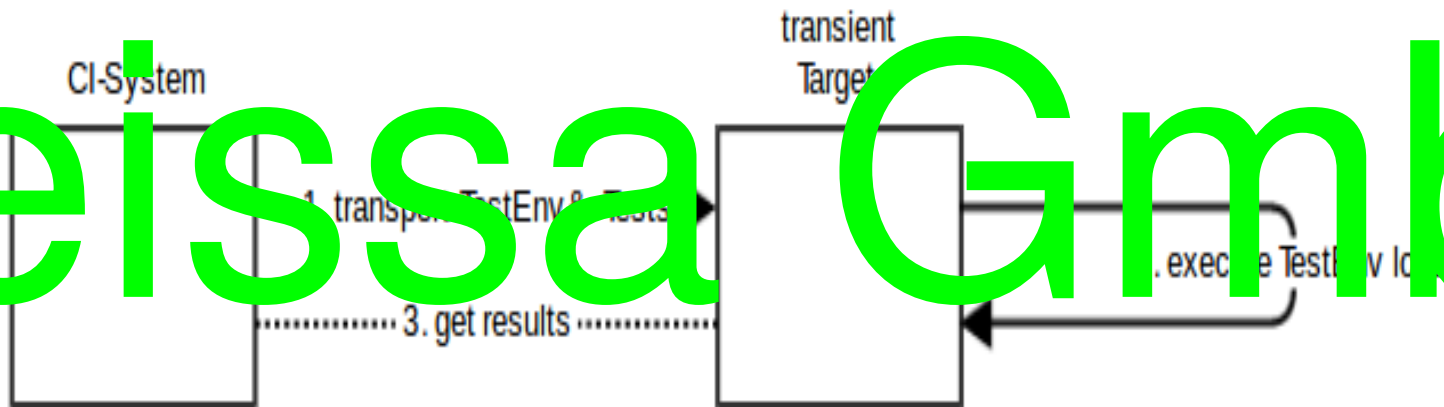


Fkt. Integration testen



Integrationstest

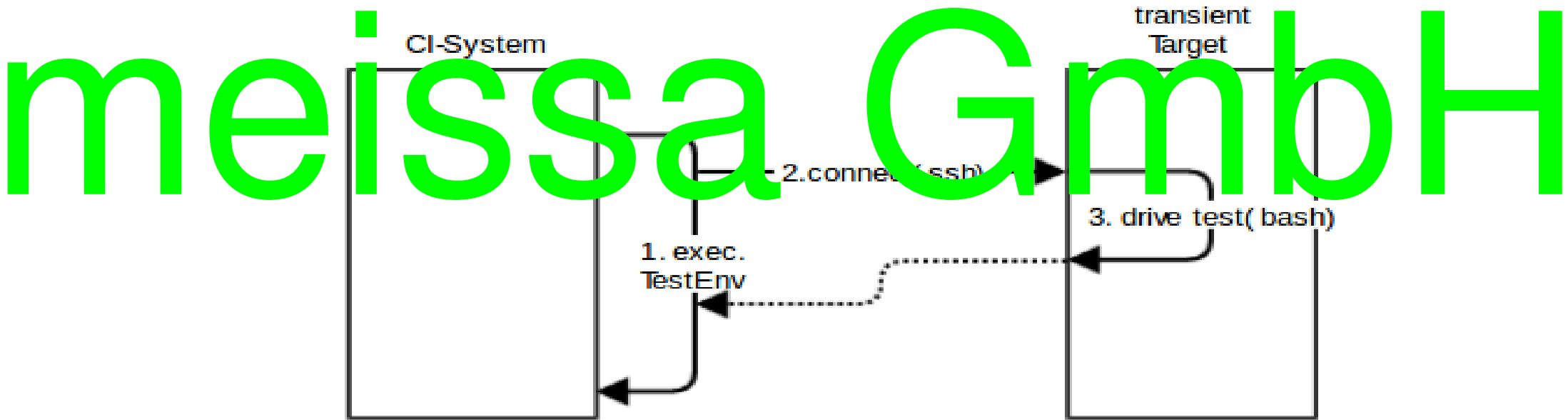
- Whitebox, controll & execute on target



meissana GmbH

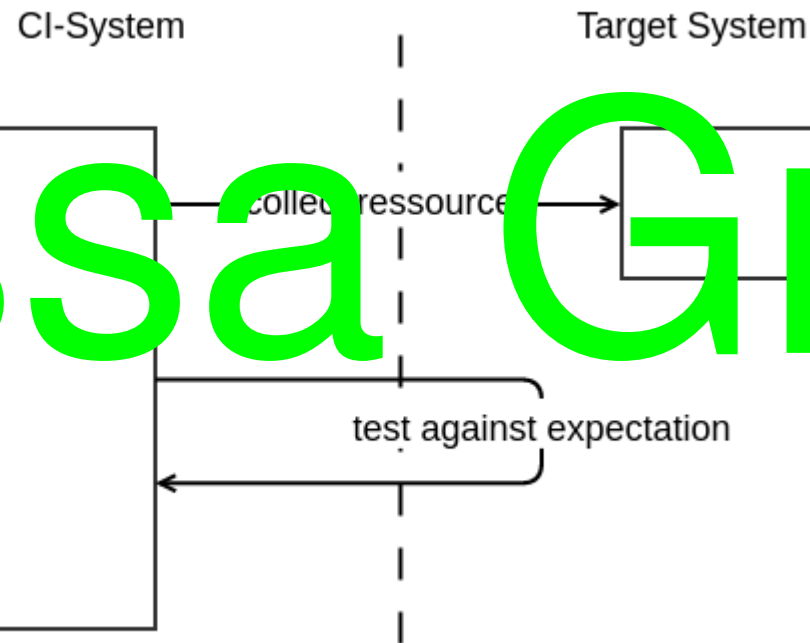
Integrationstest

- Whitebox controlled remote, executed on target



Integrationstest

- Whitebox collect facts on target

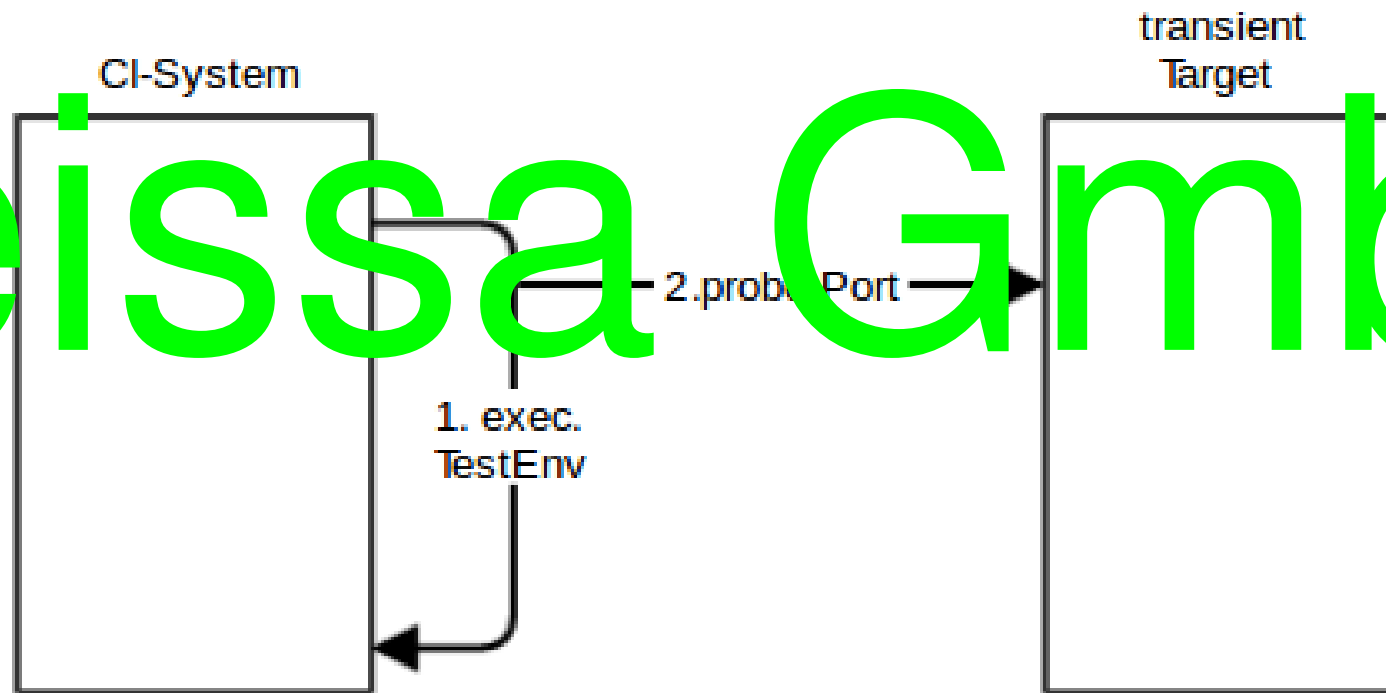


meissana GmbH



Integrationstest

- Blackbox



ServerTest-ServerSpec-Goss

```
{:netstat  
  {:apache2 {:port "80"}}
```

```
describe port(80), do  
  it { should be_listening }  
end
```

```
port:  
  tcp:80:  
    listening: true
```



Agenda

- Vorstellung
- Tests in der Softwareentwicklung

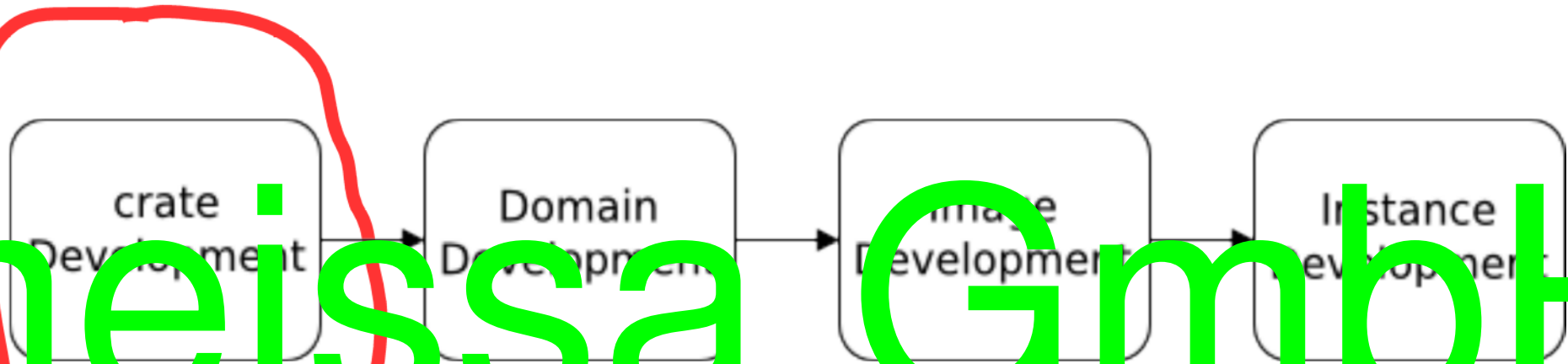
Herausforderung Operations

- Ansatzpunkte für agile DevOps

- **Testen mit dda-pallet**



Recap: Test Kontext



Infrastruktur: UnitTest File

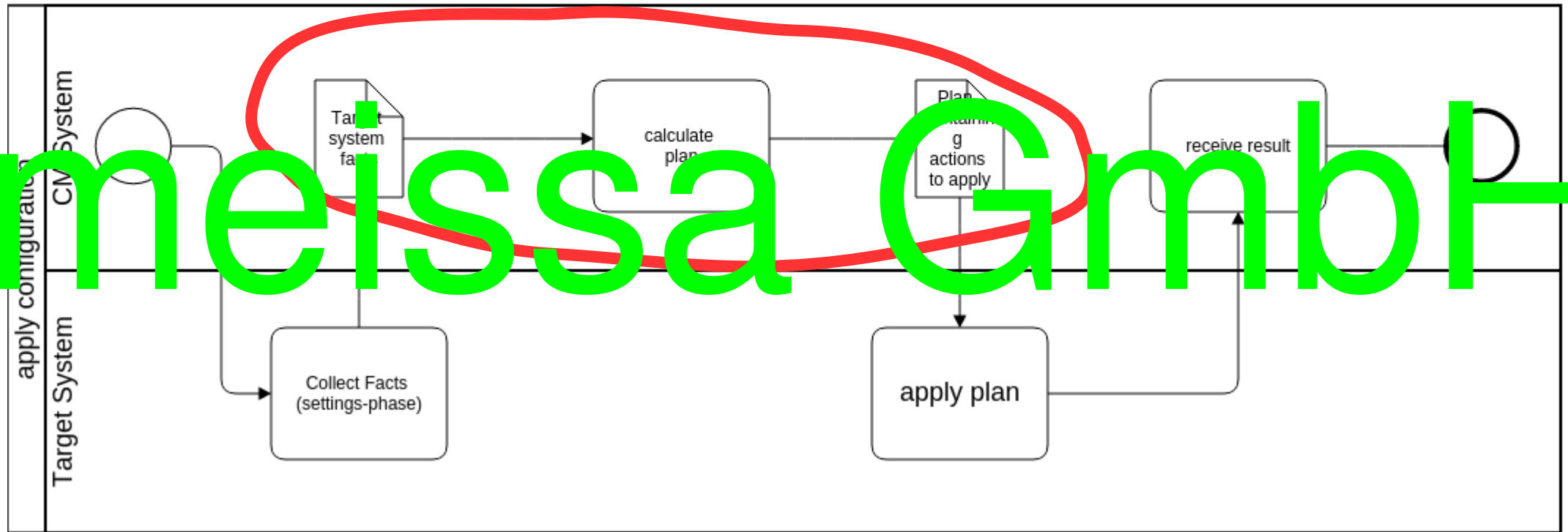
```
(vhost/vhost-conf-default-redirect-to-https-only  
:domain-name "meissa-gmbh.de"  
:server-admin-email "admin@meissa-gmbh.de")
```

REPI →

```
=> (vhost/vhost-conf-default-redirect-to-https-only  
:domain-name "meissa-gmbh.de"  
:server-admin-email "admin@meissa-gmbh.de")  
["<VirtualHost *:80>"  
 " ServerName meissa-gmbh.de"  
 " ServerAdmin admin@meissa-gmbh.de"  
 "  
 " ErrorLog \"/var/log/apache2/error.log\""  
 " LogLevel warn"  
 " CustomLog \"/var/log/apache2/access.log" combined"  
 "  
 " RewriteEngine on  
 " RewriteCond %{HTTPS} !on"  
 " RewriteRule ^/(.*)$ https://SERVER_NAME/$1  
 [R=301,L]"  
 "  
 "</VirtualHost>"]
```

meissa GmbH

Recap: Plan



Infrastruktur: UnitTest Plan



```
clojure-inspector
{:actions ([:context nil, :action-symbol pallet.actions/group, :action {:action-symbol pallet.actions/group, :execution :in-sequence, :precedence {}}, :args ("sudo" :action :create), :form (pallet.actions/gr...
[:actions ([:context nil, :action-symbol pallet.actions/group, :action {:action-symbol pallet.actions/group, :execution :in-sequence, :precedence {}}, :args ("sudo" :action :create), :form (pallet.actio...
[:session {:results ([:target {:node mock-group :ubuntu true public: 0.0.0.0 private: null mock-node-0-0-0-0, :image {:os-family :ubuntu}, :default-phases [:configure], :group-name :mock-group, :phases...
[:results ([:target {:node mock-group :ubuntu true public: 0.0.0.0 private: null mock-node-0-0-0-0, :image {:os-family :ubuntu}, :default-phases [:configure], :group-name :mock-group, :phases...
[:target {:node mock-group :ubuntu true public: 0.0.0.0 private: null mock-node-0-0-0-0, :image {:os-family :ubuntu}, :default-phases [:configure], :group-name :mock-group, :phases {:ph...
[:target-type nil]
[:plan-state {:pallet.action/action-options {}}]
[:result ([:context nil, :action-symbol pallet.actions/group, :action {:action-symbol pallet.actions/group, :execution :in-sequence, :precedence {}}, :args ("sudo" :action :create), :form (pallet...
[:context nil, :action-symbol pallet.actions/group, :action {:action-symbol pallet.actions/group, :execution :in-sequence, :precedence {}}, :args ("sudo" :action :create), :form (pallet.actio...
[:args ("pallet" :password nil :create-home true :action :create :groups ["sudo"] :shell :bash)), :always-after #{#<actions$eval18051$group__18053 pallet.actions$eval18051$group__18...
[:args (("pallet" :password nil :create-home true :action :create :groups ["sudo"] :shell :bash)), :always-after #{#<actions$eval18051$group__18053 pallet.actions$eval18051$group__18...
[:context nil, :action-symbol pallet.actions/user, :execution :aggregated, :precedence {:always-after #{#<actions$eval18051$group__18053 pallet.actions$eval18051$group__18053 pallet.actions$eval18051$group__18053...
[:form (pallet.actions/user ("pallet" :password nil :create-home true :action :create :groups ["sudo"] :shell :bash))]
[:location :target]
[:context nil, :action-symbol pallet.actions-impl/remote-file-action, :action {:action-symbol pallet.actions-impl/remote-file-action, :execution :in-sequence, :precedence {}}, :args ("/root/.ssh...
[:context nil, :action-symbol clj-action35975, :action {:action-symbol clj-action35975, :execution :in-sequence, :precedence {}}, :args (), :form (clj-action35975), :script #<init$init$action...
[:context nil, :action-symbol pallet.actions/directory, :action {:action-symbol pallet.actions/directory, :execution :in-sequence, :precedence {}}, :args ("/home/pallet/.ssh/" :mode "755" :ow...
[:context nil, :action-symbol pallet.actions-impl/remote-file-action, :action {:action-symbol pallet.actions-impl/remote-file-action, :execution :in-sequence, :precedence {}}, :args ("/home/p...
[:context nil]
[:action-symbol pallet.actions-impl/remote-file-action]
[:action {:action-symbol pallet.actions-impl/remote-file-action, :execution :in-sequence, :precedence {}}]
[:args ("/home/pallet/.ssh/authorized_keys" {:install-new-files true, :overwrite-changes nil, :owner "pallet", :content "", :mode "644"})]
[:form (pallet.actions-impl/remote-file-action "/home/pallet/.ssh/authorized_keys" {:install-new-files true, :overwrite-changes nil, :owner "pallet", :content "", :mode "644"})]
[:script [{:language :bash} "echo 'remote-file /home/pallet/.ssh/authorized_keys...';\n{\n{\n# -- START pallet implementation function\nverify_md5() {\npath=$1\nstate_path=$2\nmd5_p...
[:summary "remote-file /home/pallet/.ssh/authorized_keys :content \"...\""]
[:action-type :script]
[:location :target]
[:context nil, :action-symbol pallet.actions-impl/remote-file-action, :action {:action-symbol pallet.actions-impl/remote-file-action, :execution :in-sequence, :precedence {}}, :args ("/etc/sud...
[:context nil, :action-symbol pallet.actions/directory, :action {:action-symbol pallet.actions/directory, :execution :in-sequence, :precedence {}}, :args ("/home/pallet/state" :group "pallet" :...]
```

meissa GambH

Integration Test

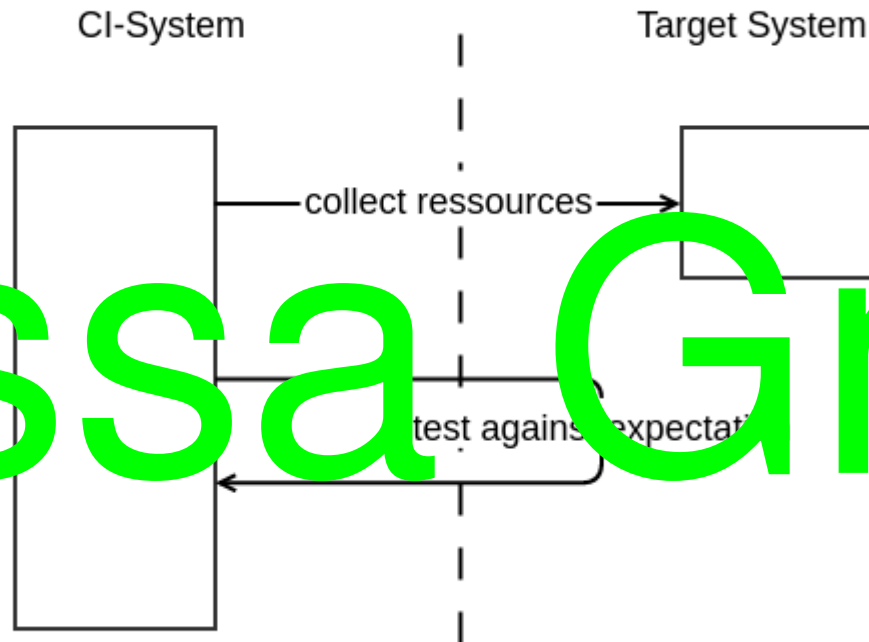
```
(def default-vhost-config
  {:domain-name "localhost.localdomain"
   :listening-port "443"
   :server-admin-email "admin@localdomain"
   ...
  })
```

meissaa GmbH

```
{:netstat
  {:apache2 {:port "443"}}}
```



Integration Test dda-servertest-crate



meissana GmbH



Resource on target

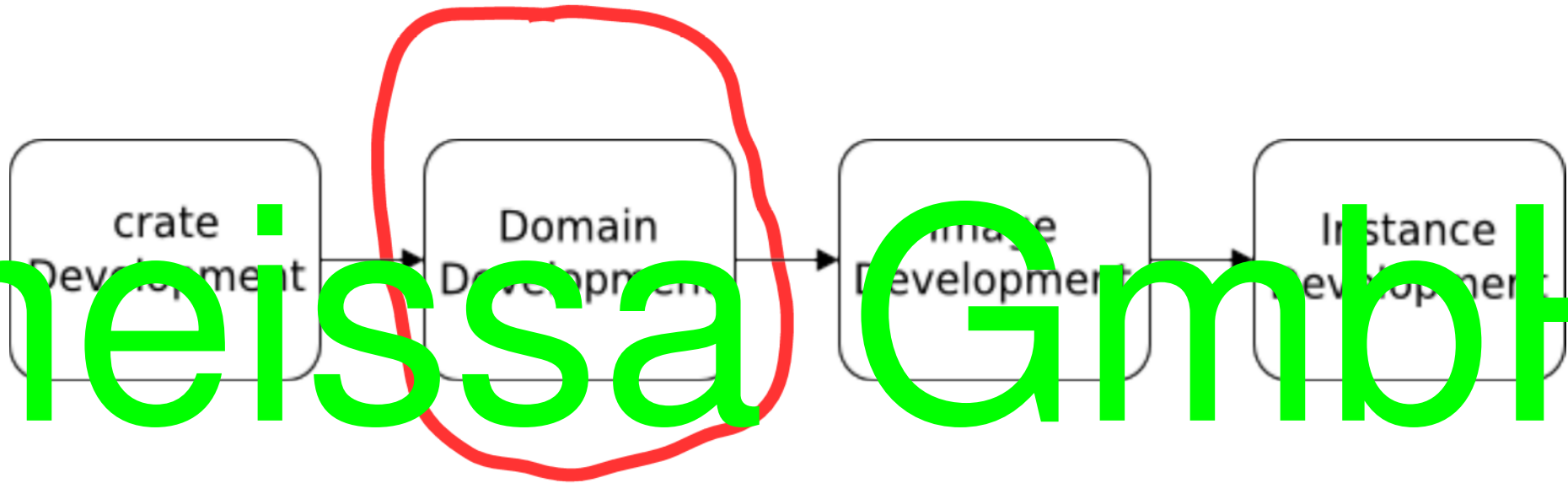
```
(defn collect-netstat-fact []  
  (collect-fact fact-id-netstat  
    ("netstat" "-tulpen")  
    :transform-fn parse-netstat))
```

meissa GmbH

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	User	Inode	PID/Program name
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	0	9807	1001/sshd
tcp6	0	0	:::80	:::*	LISTEN	0	44161	4135/apache2
tcp6	0	0	:::4369	:::*	LISTEN	108	33687	27416/epmd



Recap: Test Kontext



Domain: Generate Config

```
(def archiva-config  
  {:fqdn "artifacts.meissa-gmbh.de"  
   :smtp-mail-host "mail-server"  
   :smtp-mail-user "mail-account"  
   :smtp-mail-password "mail-pwd"})
```

```
(sut/archiva-httpd-config archiva-config)
```

REPL →

```
=> (sut/archiva-httpd-config archiva-config)  
{:limits  
  {:server-limit 150, :max-clients 150},  
  :jk-configuration  
  {:jkStripSession "On",  
   :jkWatchdogInterval 120},  
  :vhosts  
  {:default  
   [[:domain-name  
     "artifacts.meissa-gmbh.de",  
     :listening-ports [43],  
     :server-admin-email  
     "admin@artifacts.meissa-gmbh.de",  
     :maint-notice-page-content  
     ["<h1>Webserver Maintenance Mode</h1>"],  
     :maint-notice-url  
     "http://artifacts.meissa-gmbh.de"],  
     :port "8009",  
     :host "127.0.0.1",  
     :worker "mod_jk_www",  
     :maintain-timeout-sec 90,  
     :socket-connect-timeout-ms 62000},  
     :cert-letsencrypt  
     {:letsencrypt-mail  
      "admin@artifacts.meissa-gmbh.de"}]}}
```

meissa GmbH

Domain: Unit Test

```
(deftest test-httpd-vhost []  
  (testing  
    "httpd vhost domain test"  
    (is (=   
      {:limits ...,  
       :vhosts  
       {:default  
        {:domain-name  
         "artifacts.meissa-gmbh.de",  
         ...},  
        :cert-letsencrypt  
        {:letsencrypt-mail  
         "admin@artifacts.meissa-gmbh.de"}}}})  
      (/archiva httpd-config archiva-config  
        ))  
      ))  
    (run-tests)
```

REPL



```
=> (run-tests)  
  
Testing de.meissa-gmbh.pallet.crate.archiva.app-test  
  
Ran 1 tests containing 1 assertions.  
0 failures, 0 errors.  
{:test 1,  
 :pass 1,  
 :fail 0,  
 :error 0,  
 :type :success
```

meiSSa GmbH



Boundaries with Validation

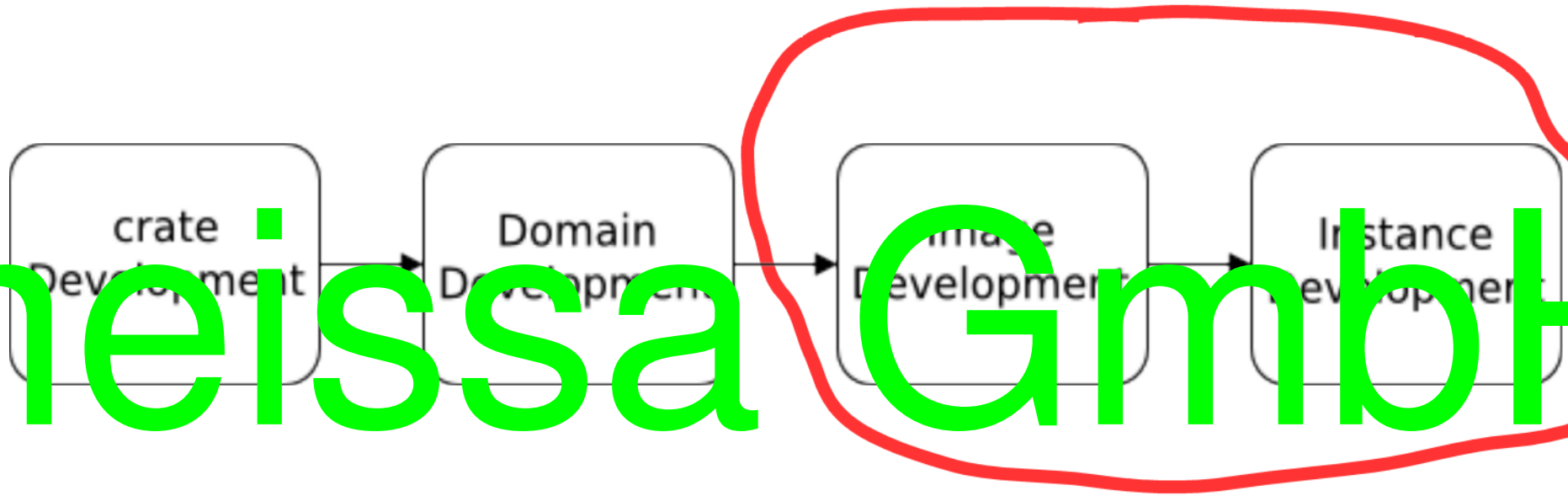
```
(def VhostConfig
  "defines a schema for a httpdConfig"
  {:domain-name s/Str
   :listening-port s/Str
   :server-admin-email s/Str
   :access-control [s/Str]
   (s/optional-key :document-root) s/Str
   (s/optional-key :rewrite-les) [s/Str]
   (s/optional-key :user-credentials) [s/Str]
   (s/optional-key :aliases) [(url s/Str path s/Str)]
   (s/optional-key :location) [(s/optional-key :basic-auth) s/Bool
                                (s/optional-key :locations-override) [s/Str]
                                (s/optional-key :path) s/Str]
   (either :letsencrypt-or-manual-certificate
    (s/optional-key :cert-letsencrypt) s/Str
    (s/optional-key :cert-manual) s/Str)
   :domains [s/Str]
```

meissa GmbH

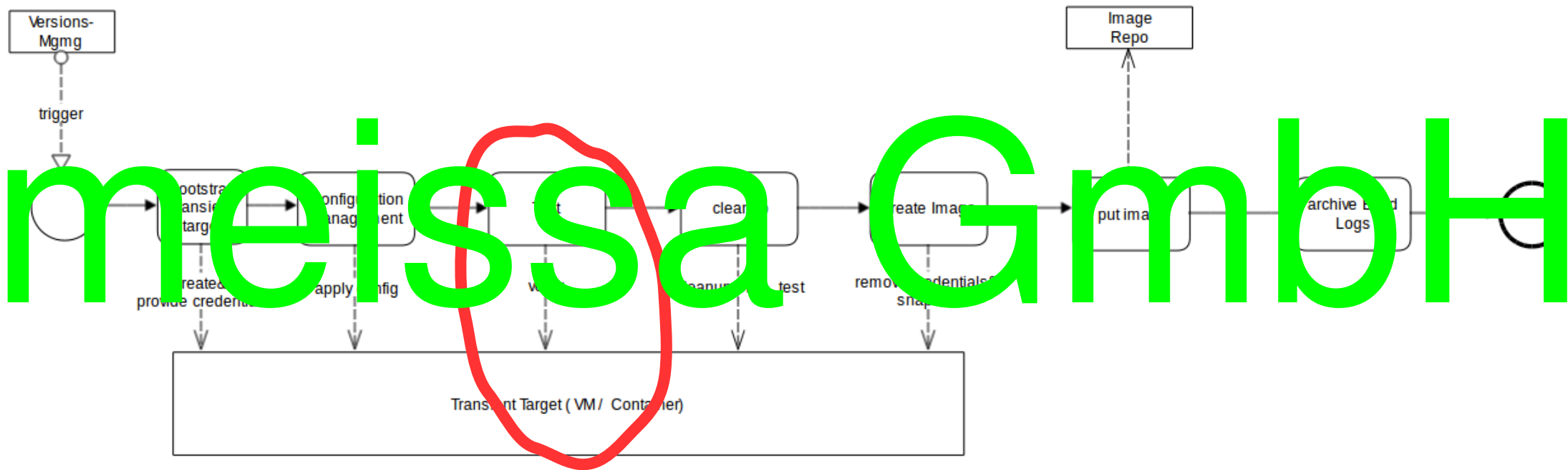
```
(s/defn ^:always-validate configure-httpd
  [config :- HttpdCrateConfig]
  (server/configure config)
  (vhost/configure config))
```



Recap: Test Kontext



Fkt. Integration testen



Smoke-Test Image / Instance

```
{:package
  {:apache2 {:installed true}}
 :netstat
  {:apache2 {:port "443"}}
 :file
  {:dpkg-lock
   {:path "/var/lib/dpkg/lock"
    :exist? false}}}}
```

meiSSa GmbH



Fazit

- Respekt vor den Unterschieden
- Von der Softwareentwicklung lernen

meiSSa GmbH

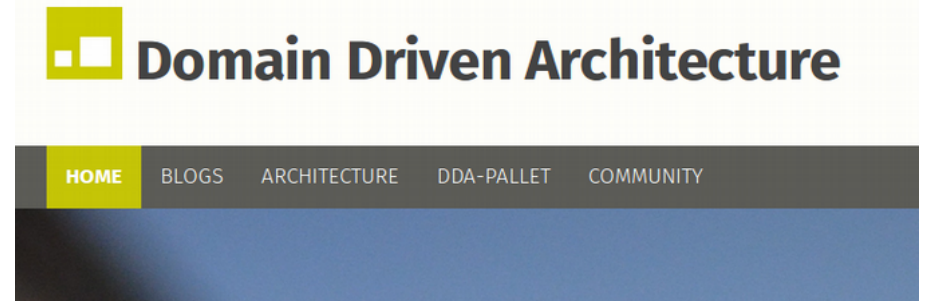
• Effizientes DevOps Testen möglich!



Kontakt

Informiert bleiben - Blog:

<https://DomainDrivenArchitecture.org>



meiessa GmbH

Kennzeichen: meietop

<https://www.meietop.com/de/dda-pallet-DevOps-Hacking-with-Cojur>



Ausprobieren & Unterstützung - GitHub

<https://github.com/DomainDrivenArchitecture/dda-pallet-masterbuild>

DomainDrivenArchitecture / [dda-pallet-masterbuild](#)