



DCAE - Service Components Onboarding and Lifecycle Management

Vijay VK(vv770d@att.com) & AT&T DCAE Team
27 March 2018 (ONS)

AGENDA

Platform Architecture

Service Component Pre-requisite

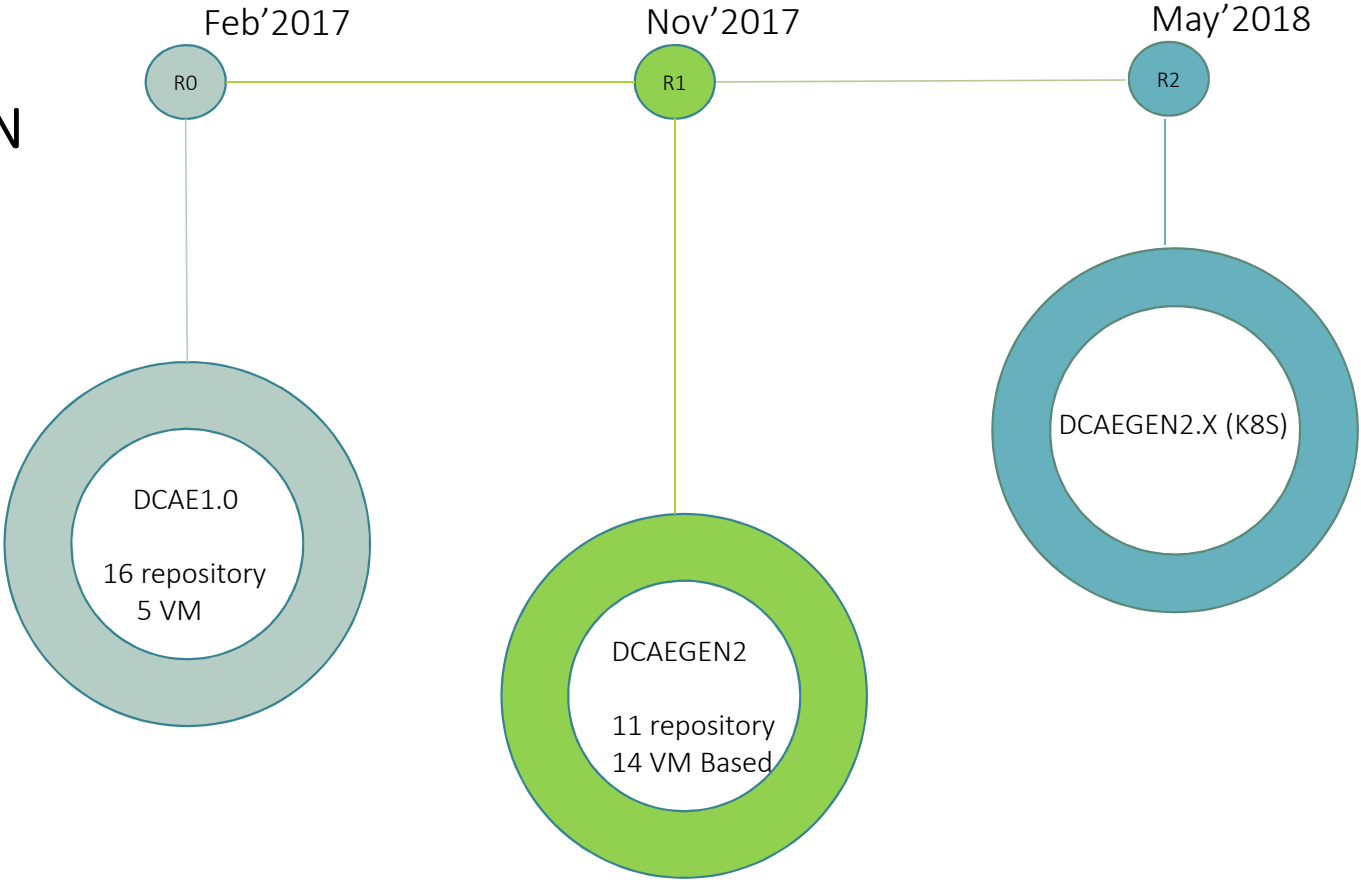
Onboarding Flow

Catalog & Service Template

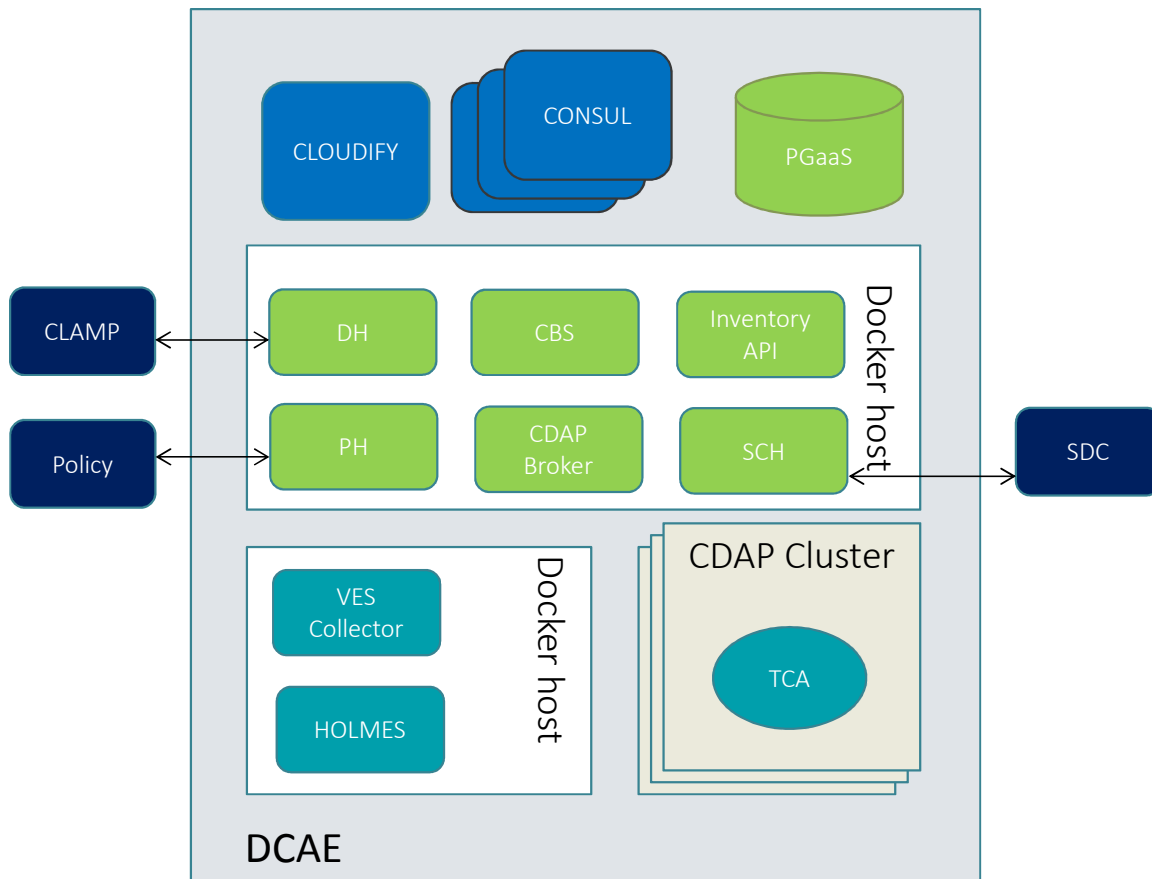
Control Loop Configuration Flow

R3 Focus

DCAE PLATFORM EVOLUTION



ONAP DCAE Architecture (R1)

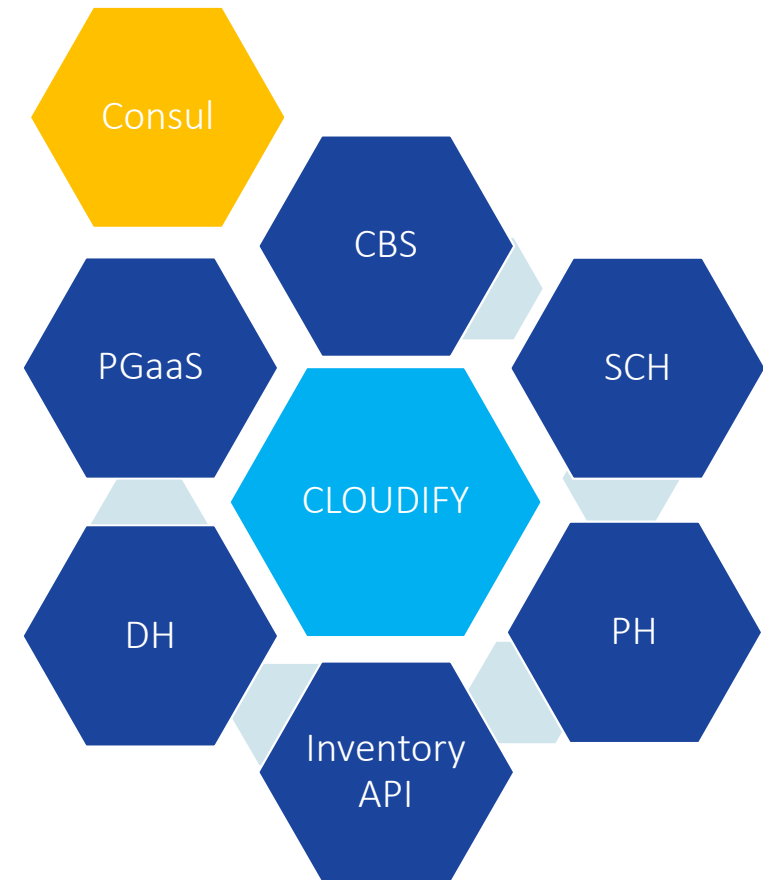


DCAE Platform components

- Cloudify (& Plugins)
- Consul
- Config Binding Service (CBS)
- Service change Handler (SCH)
- Policy Handler (PH)
- Deployment Handler (DH)
- Inventory API
- CDAP Broker
- PGaaS
- CDAP Cluster
- Docker Hosts

R2 Platform Updates

- Utilizing OOM Kubernetes cluster
- No Hadoop cluster deployment
- Containerized Cloudify
- Using Helm chart for Bootstrap, Cloudify Manager, Redis
- All platform (and services) deployment through Cloudify (custom K8 plugin)
- PGaaS deployment through OOM and Cloudify
- OOM Deployed Consul will be utilized
- MSB (Api and websocket)
- Policy Flow enablement



Modular

Well-defined interfaces

Light weight

Scale quickly

Technology freedom

Deployment Flexibility

Service
Components

Service Component Requirements

Overall functionality

Spec creation and
validation through
dcae_cli utility

State
synchronization

Utilize
ConfigBindingService
API's

Enforcement of
Policy Decisions

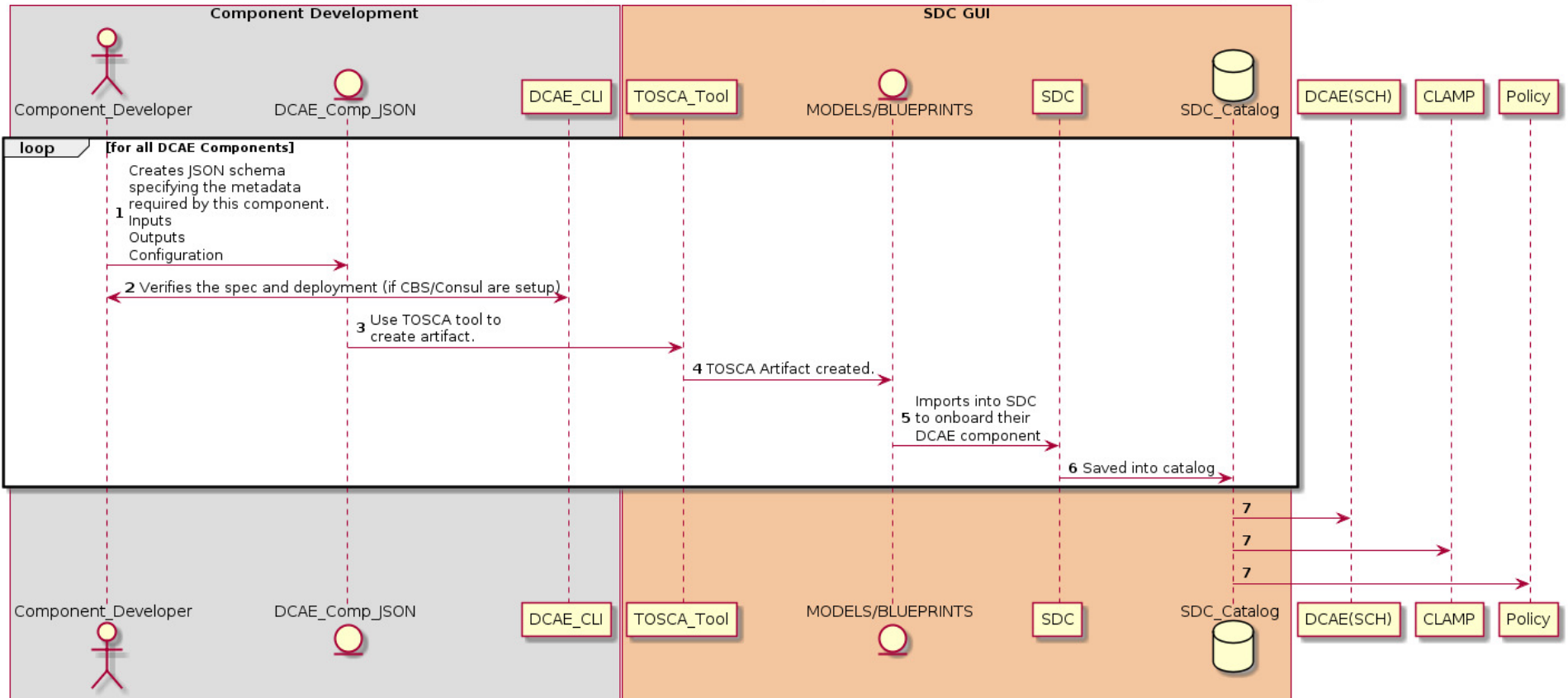
Design time requirement

Component specification

- Defining input and output structure
- Topic Keys
- Configuration
 - ✓ Design-time
 - ✓ Deployment-time
 - ✓ Policy editable
- Policy Model
- Filesystem mapping
- Port mapping
- Image specification

Onboarding Flow (Target)

This is the flow for onboarding Service components. All components get onboarded and made available to SDC template designer.



Component spec (Example)

```
1 {
2   "self": {
3     "version": "1.2.1",
4     "name": "dcae-ves-collector",
5     "description": "Collector for receiving VES events through restful interface",
6     "component_type": "docker"
7   },
8   "streams": {
9     "subscribes": [],
10    "publishes": [
11      {
12        "format": "VES_specification",
13        "version": "5.28.4",
14        "type": "message router",
15        "config_key": "ves-fault"
16      }
17    ],
18   },
19   "parameters": [
20     {
21       "name": "collector.service.port",
22       "value": -1,
23       "description": "standard http port"
24     },
25     {
26       "name": "collector.service.secure.port",
27       "value": 8443,
28       "description": "secure port ",
29       "designer_editable": true
30     }
31   ],
32   "auxiliary": {
33     "healthcheck": {
34       "type": "https",
35       "interval": "15s",
36       "timeout": "1s",
37       "endpoint": "/healthcheck"
38     },
39     "volumes": [
40       {
41         "container": {
42           "bind": "/opt/app/dcae-certificate"
43         },
44         "host": {
45           "path": "/opt/app/dcae-certificate"
46         }
47       },
48       {
49         "container": {
50           "bind": "/opt/app/VESCollector/Logs"
51         },
52         "host": {
53           "path": "/opt/Logs/DCAE/VESCollector/Logs"
54         }
55       }
56     ],
57     "ports": [
58       "8443:8443"
59     ]
60   },
61   "artifacts": [
62     {
63       "type": "docker image",
64       "uri": "nexus.onap.org:10001/onap/org.onap.dcaeagen2.collectors.ves.vescollect
65     }
66   ]
67 }
```

Blueprint (Example)

```
18 tosca_definitions_version: cloudfify_dsl_1_3
19
20 description: >
21 This blueprint will install the ves collector and provision the needed message router topics.
22
23 imports:
24 - http://www.getcloudfify.org/spec/cloudfify/3.4/types.yaml
25 - https://NEXUS_REPO_HOST:8443/repository/NEXUS_RAW/type_files/docker/2.2.0/node-type.yaml
26 - https://NEXUS_REPO_HOST:8443/repository/NEXUS_RAW/type_files/relationship/1.0.0/node-type.y
27 - http://NEXUS_REPO_HOST:8081/repository/NEXUS_RAW/type_files/dmaap/dmaap_mr.yaml
28
29 inputs:
30
31 service_id:
32   description: Unique id used for an instance of this DCAE service. Use deployment id
33   default: 'foobar'
34 location_id:
35   default: 'foocentral'
36 docker_host_override:
37   default: 'component_dockerhost'
38
39 topic00_aaf_username:
40 topic00_aaf_password:
41 topic00_location:
42   default: mtc5
43 topic00_client_role:
44   default: com.att.dcae.member
45
46
47 node_templates:
48 topic00:
49   type: dcae.nodes.Topic
50   properties:
51     topic_name: sec-fault-unsecure
52 topic01:
53   type: dcae.nodes.Topic
54   properties:
55     topic_name: sec-measurement-unsecure
56 topic02:
57   type: dcae.nodes.Topic
58   properties:
59     topic_name: sec-heartbeat-unsecure
60 topic03:
61   type: dcae.nodes.Topic
62   properties:
63     topic_name: sec-other-unsecure
64 component00:
65   type: dcae.nodes.DockerContainerForComponentsUsingDmaap
```

```
86 properties:
87   service_component_type:
88     'dcae-controller-ves-collector'
89   service_id:
90     { get_input: service_id }
91   location_id:
92     { get_input: location_id }
93   application_config:
94     collector.keystore.passwordfile: "/opt/app/dcae-certificate/.password"
95     collector.service.secure.port: -1
96     tomcat.maxthreads: '200'
97     collector.keystore.file.location: "/opt/app/dcae-certificate/keystore.jks"
98     header.authflag: 0
99     collector.service.port: 8080
100     streams_publishes:
101       sec_fault_unsecure:
102         aaf_password: { get_input: topic00_aaf_password }
103         dmaap_info: "<<topic00>>"
104
105     services_calls: {}
106     collector.schema.checkflag: 1
107     collector.dmaap.streamid: fault=sec_fault,roadm-sec-to-hp|syslog=sec_syslog|heartbeat=sec_heartbeat|
108     header.authlist: userid1,base64encodepwd1|userid2,base64encodepwd2
109     streams_subscribes: {}
110     collector.inputQueue.maxPending: 8096
111     collector.schema.file: "./etc/CommonEventFormat_27.2.json"
112     collector.keystore.alias: dynamically generated
113 image:
114   NEXUS_REPO_HOST:18443/dcae-dev-raw/dcae-controller-ves-collector:1.1
115 docker_config:
116   healthcheck:
117     type: "http"
118     interval: "15s"
119     timeout: "1s"
120     endpoint: "/"
121   ports:
122     - 8443:8443
123   volumes:
124     - container:
125       bind: /opt/app/dcae-certificate
126       host:
127         path: /opt/app/dcae-certificate
128     - container:
129       bind: /opt/app/VESCollector/logs
130       host:
131         path: /opt/logs/DCAE/VESCollector/logs
132   streams_publishes:
133     - name: topic00
134       location: { get_input: topic00_location }
135       client_role: { get_input: topic00_client_role }
136       type: message_router
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
```

VES Collector

TCA

Holmes

Heartbeat Service

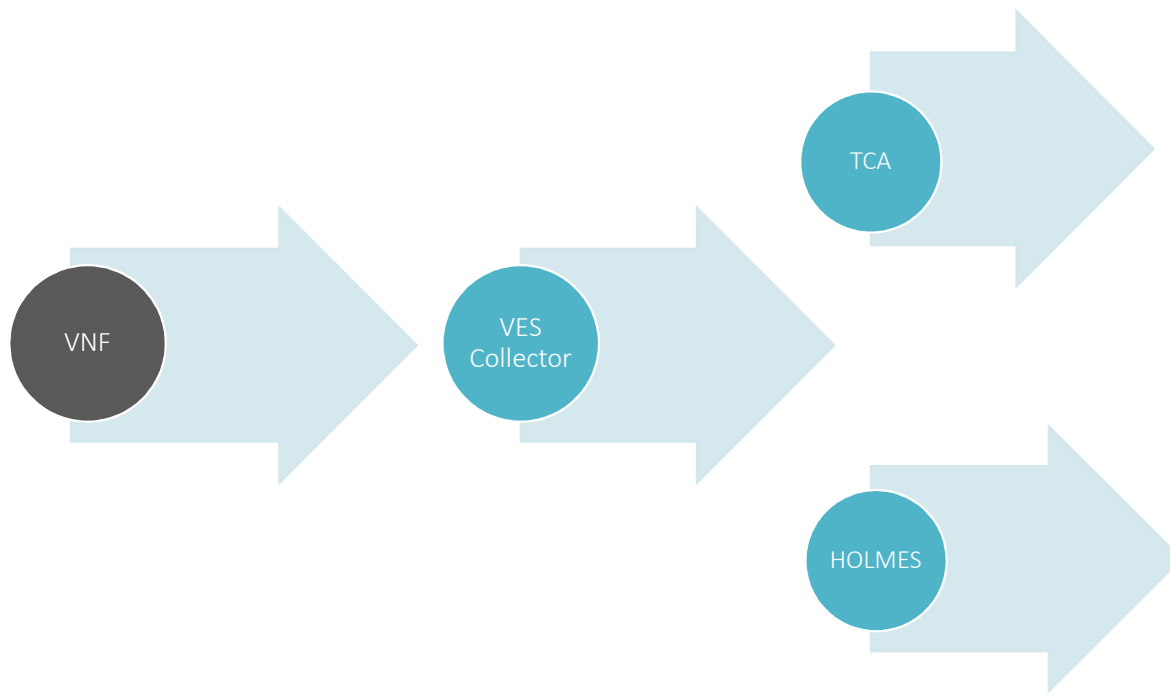
Mapper Service

SNMP Trap Collector

PRH

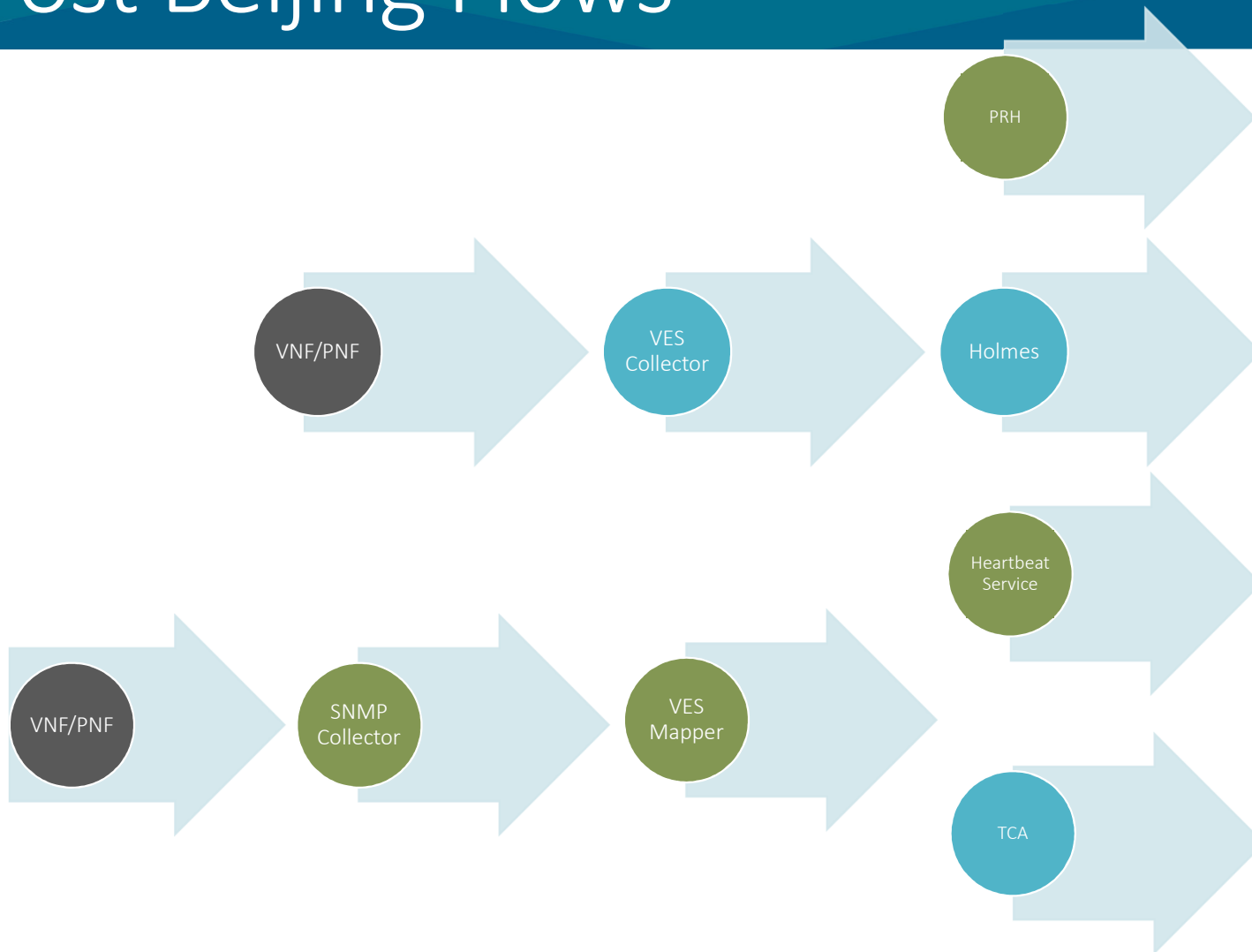
Service Component Catalog

Amsterdam Flows

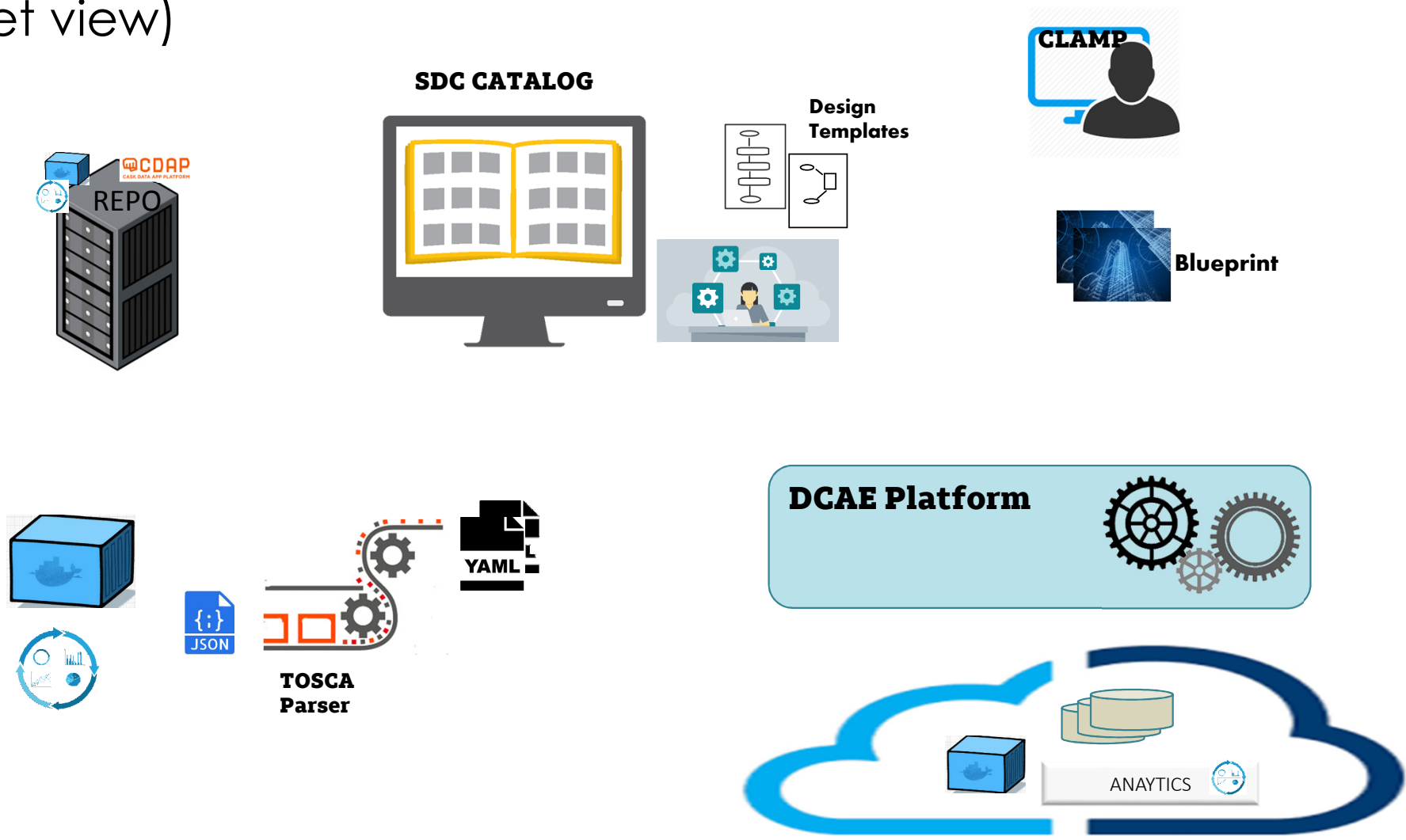


- Virtual Firewall
- Virtual DNS
- vCPE
- VoLTE

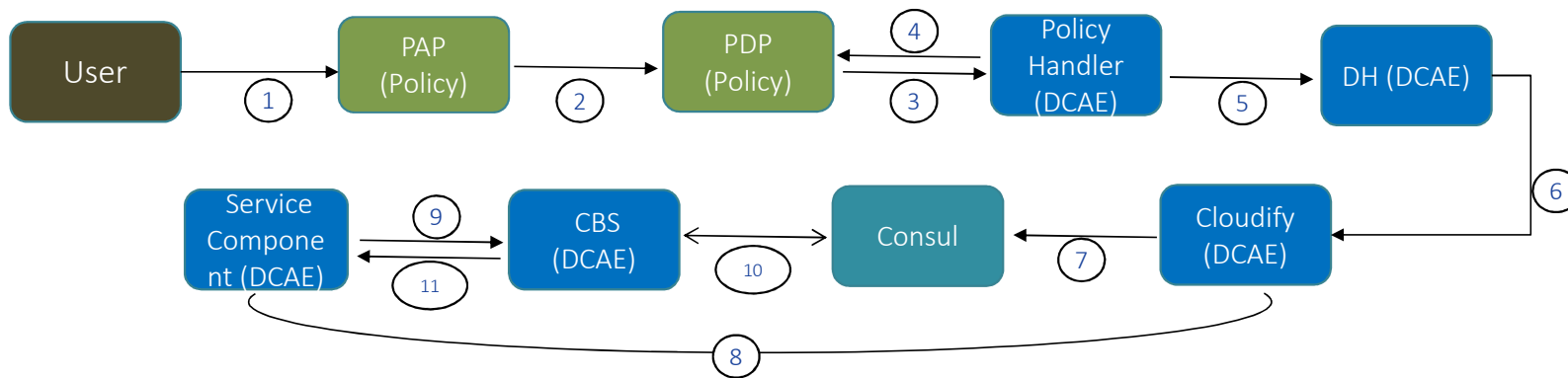
Post Beijing Flows



Onboarding, Service Design & Deployment Flow (Target view)



Configuration Flow



1 → Policy Create/update/delete

2 → Policy distribution

3 → Notification on policy change

4 → Get updated policy list

5 → If DCAE scope is updated; get latest policy config and notify deployment handler with deltas

6 → Identify components matching policy received.

7 → Update list of policy into consul KV under SCN folder

8 → Notification service component on policy update

9 → Invoke CBS API (<CBS>/service_component_all/) to fetch latest policy

10 → Fetches config under SCN folder and returns json object

11 → Receives updated policy and enforce new rules.

Policy Model/Configuration (sample)

```
policy_schema
├── 0
│   ├── name: "domain"
│   ├── description: "Domain name to which TCA needs to be applied"
│   ├── type: "string"
│   └── value: "measurementsForVfScaling"
├── constraints
│   └── 0
│       └── equal: "measurementsForVfScaling"
├── 1
│   ├── name: "metricsPerEventName"
│   ├── description: "Contains eventName and threshold details that need to be applied to given eventName"
│   ├── type: "list"
│   └── entry_schema
│       ├── 0
│       │   ├── name: "eventName"
│       │   ├── description: "Event name to which thresholds need to be applied"
│       │   ├── type: "string"
│       │   └── value: ""
│       ├── 1
│       │   ├── name: "controlLoopSchemaType"
│       │   ├── description: "Specifies Control Loop Schema Type for the event Name e.g. VNF, VM"
│       │   ├── type: "string"
│       │   └── value: ""
│       ├── constraints
│       │   └── 2
│       │       ├── name: "policyScope"
│       │       ├── description: "TCA Policy Scope"
│       │       ├── type: "string"
│       │       └── value: ""
│       ├── 3
│       │   ├── name: "policyName"
│       │   ├── description: "TCA Policy Scope Name"
│       │   ├── type: "string"
│       │   └── value: ""
│       ├── 4
│       │   ├── name: "policyVersion"
│       │   ├── description: "TCA Policy Scope Version"
│       │   ├── type: "string"
│       │   └── value: ""
│       └── 5
│           ├── name: "thresholds"
│           ├── description: "Thresholds associated with eventName"
│           ├── type: "list"
│           └── entry_schema
```

spec

```
[tosca_definitions_version: tosca_simple_yaml_1_0_UU.....]
data_types:
  policy_data_metricsPerEventName:
    ies:
      allLoopSchemaType:
        s: string
        r: Specifies Control Loop Schema Type for the event Name e.g. VNF, VM
        straits:
          allid_values:
            VM
            VNF
          Name:
            s: string
            r: Event name to which thresholds need to be applied
          yName:
            s: string
            r: TCA Policy Scope Name
          yScope:
            s: string
            r: TCA Policy Scope
          yVersion:
            s: string
            r: TCA Policy Scope Version
        solids:
          s: list
          r: Thresholds associated with eventName
        ry_schema:
          ype: policy_data.thresholds
          ta_tca_policy:
            ies:
              1:
                s: string
                r: Domain name to which TCA needs to be applied
                straits:
                  qual: measurementsForVfScaling
                csPerEventName:
                  s: list
                  r: Contains eventName and threshold details that need to be applied to given eventName
                ry_schema:
                  ype: policy_data.metricsPerEventName
                  ta_thresholds:
                    ies:
                      allLoopControlName:
                        s: string
                        r: Closed Loop Control Name associated with the threshold
                      allLoopEventStatus:
                        s: string
                        r: Closed Loop Event Status of the threshold
                      straits:
                        allid_values:
                          ONSET
                          ABATED
                        tion:
                          s: string
                          r: Direction of the threshold
                        straits:
                          allid_values:
                            LESS
                            LESS_OR_EQUAL
                            GREATER
                            GREATER_OR_EQUAL
                            EQUAL
                        Path:
                          s: string
                          r: Json field Path as per CEF message which needs to be analyzed for TCA
                        ity:
                          s: string
                          r: Threshold Event Severity
                    description: Threshold Event Severity
                    constraints:
```

Tosca model

```
{
  "domain": "measurementsForVfScaling",
  "metricsPerEventName": [
    {
      "eventName": "Mfvs_eNodeB_RANKPI",
      "controlLoopSchemaType": "VNF",
      "policyScope": "resource=vFirewall;type=configuration",
      "policyName": "configuration.dcae.microservice.tca.xml",
      "policyVersion": "v0.0.1",
      "thresholds": [
        {
          "closedLoopControlName": "CL-FRWL-LOW-TRAFFIC-SIG-d9",
          "closedLoopEventStatus": "ONSET",
          "version": "1.0.2",
          "fieldPath": "$.event.measurementsForVfScalingFields",
          "thresholdValue": 4000,
          "direction": "LESS_OR_EQUAL",
          "severity": "MAJOR"
        },
        {
          "closedLoopControlName": "CL-FRWL-HIGH-TRAFFIC-SIG-E",
          "closedLoopEventStatus": "ONSET",
          "version": "1.0.2",
          "fieldPath": "$.event.measurementsForVfScalingFields",
          "thresholdValue": 20000,
          "direction": "GREATER_OR_EQUAL",
          "severity": "CRITICAL"
        },
        {
          "closedLoopControlName": "CL-FRWL-HIGH-TRAFFIC-SIG-E",
          "closedLoopEventStatus": "ABATED",
          "version": "1.0.2",
          "fieldPath": "$.event.measurementsForVfScalingFields",
          "thresholdValue": 0,
          "direction": "EQUAL",
          "severity": "CRITICAL"
        }
      ]
    }
  ]
}
```

Control Loop configuration

DCAE platform functions after onboarding

- Deployment of service components based operation requirement and business need.
- Retrieval of blueprint via SDC distribution and store in DCAE inventory
- Retrieval of configuration policy and policy updates from Policy and making available in Consul
- Service component deployment (and undeploy) support through CLAMP
- Registration (deregistration) of components in Consul and/or Kubernetes platform for healthcheck and application state
- Service registration (deregistration) with MSB when applicable
- DMaaP resource provisioning when applicable
- Provides configuration api and policy configuration via CBS
- Components scaling

R3 Focus Areas

- SDC Design Studio integration for onboarding and generation of K8S blueprint
- Expanding Service Component portfolio
- Service component Policy modelling
- Enhanced S3P level support
- AAF Integration
- Integration with newer Analytics Platform (PNDA/Flink)



Questions?