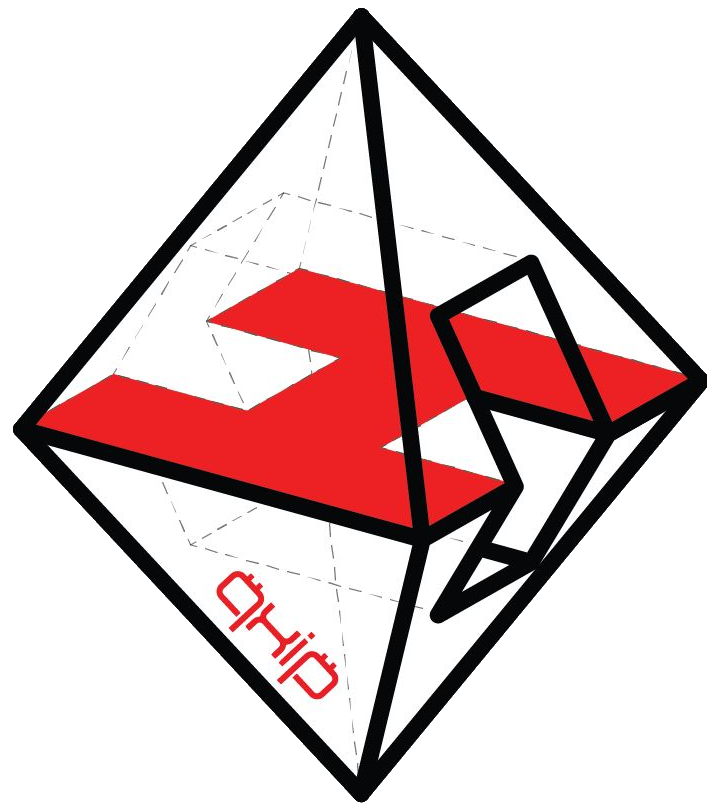


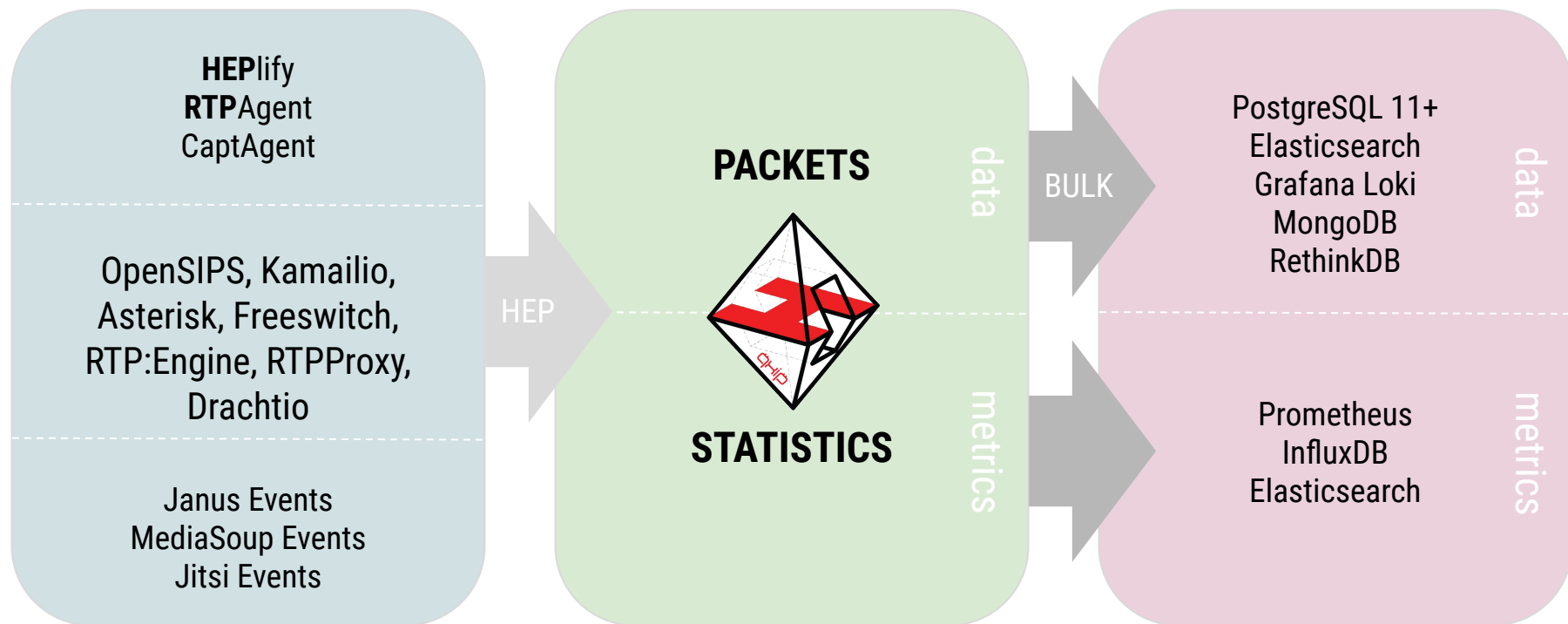
H7 WORKSHOP

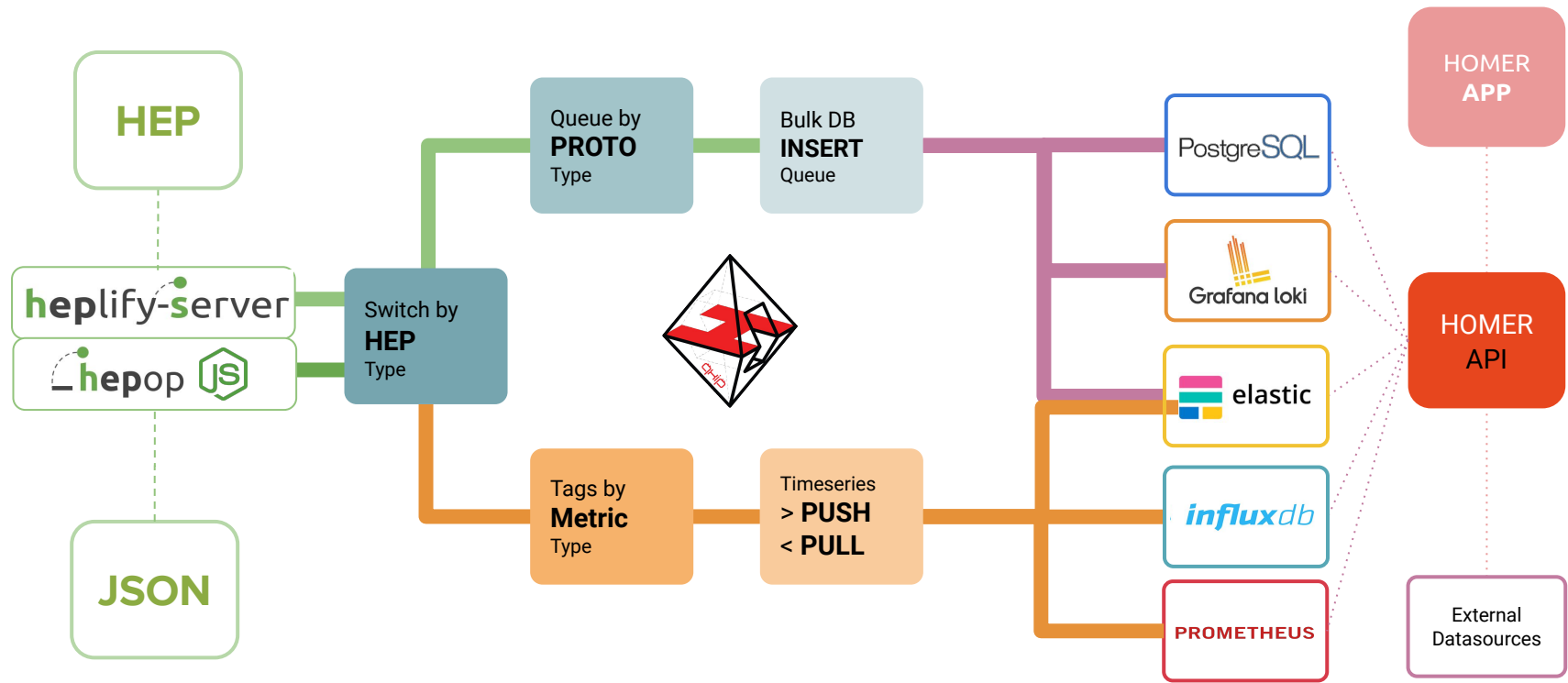


Live questions!

<https://gitter.im/sipcapture/homer>

HOMER Seven



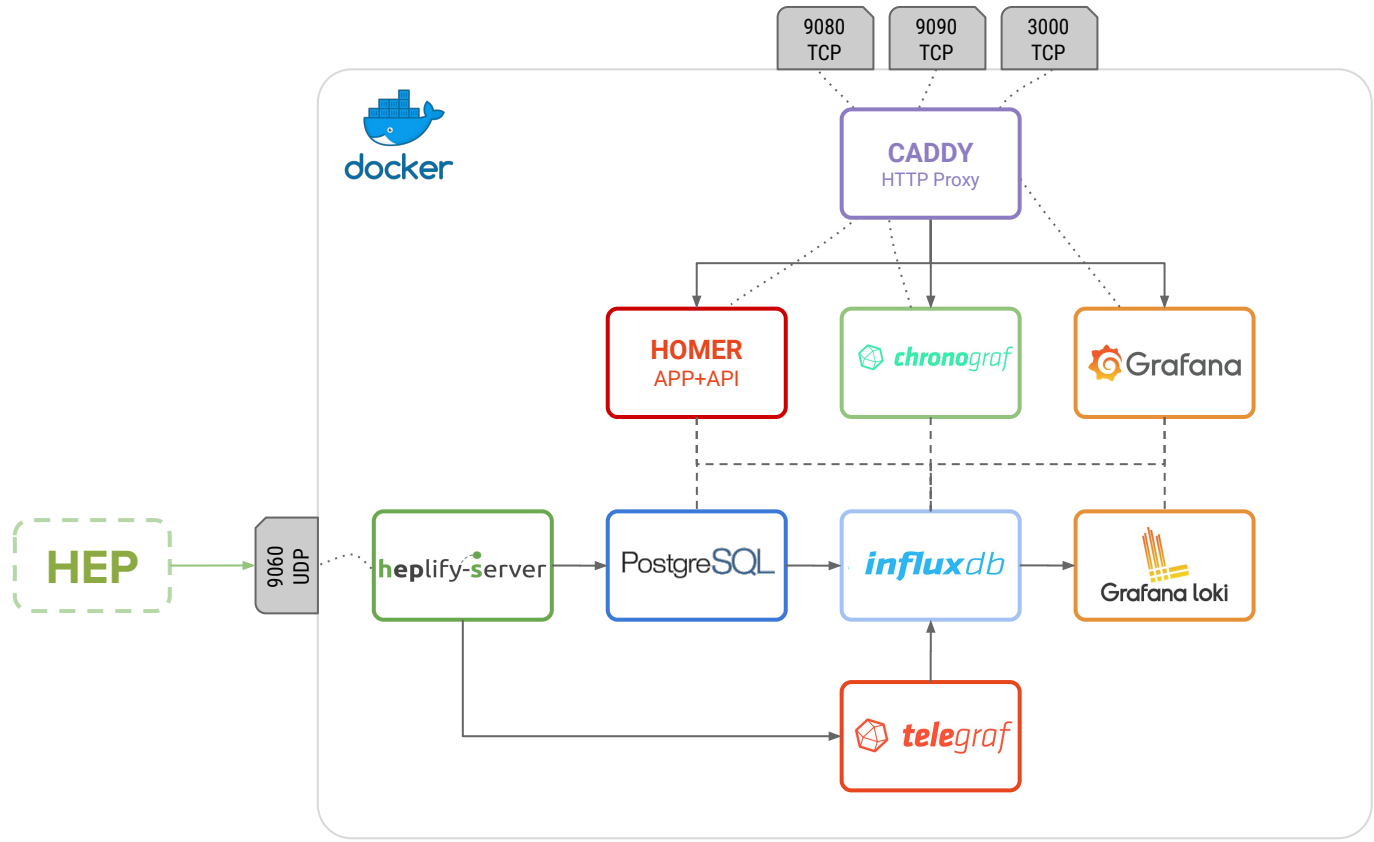


HEP FLOW

Choose your **H7** flavour

<https://github.com/sipcapture/homer7-docker>

- [HEPlify-Server](#)
 - [HEPlify + Prometheus](#)
 - [HEPlify + Prometheus + Loki](#)
 - [HEPlify + InfluxDB](#)
 - [HEPlify + InfluxDB + LoudML](#)
 - [HEPlify + Elastic](#)
 - [HEPlify + Elastic Headless](#)
- [HEPop Server](#)
 - [HEPop + InfluxDB](#)
 - [HEPop + InfluxDB + LoudML](#)
 - [HEPop + InfluxDB + JSON](#) for Janus/Mediasoup
 - [HEPop + Elasticsearch + JSON](#) for Janus/Mediasoup



WORKSHOP DOCKER FLOW

Spin-up your H7 Lab

Lab Requirements:

- linux host
 - docker-ce
 - docker-compose (latest)
- 1+ CPU, 8+ GB RAM, 20+ GB HDD
- <https://docs.docker.com/install/>
- <https://docs.docker.com/v17.09/compose/install/>

```
git clone https://github.com/sipcapture/homer7-docker --branch workshop --single-branch
cd homer7-docker/heplify-server/hom7-influx-graf-loki
docker-compose up
```


A closer look to compose file: caddy

```
caddy:
  container name: caddy
  image: stefanprodan/caddy
  container_name: caddy
  ports:
    - "3000:3000"
    - "9090:9090"
    - "9080:9080"
  volumes:
    - ./caddy/:/etc/caddy/
  environment:
    - ADMIN_USER=${ADMIN_USER:-admin}
    - ADMIN_PASSWORD=${ADMIN_PASSWORD:-sipcapture}
```

A closer look to compose file: caddy

caddy/Caddyfile

```
:9090 {
  basicauth / {$ADMIN_USER} {$ADMIN_PASSWORD}
  proxy / chronograf:8888 {
    transparent
  }
  errors stderr
  tls off
}
:9080 {
  proxy / homer-webapp:80 {
    transparent
  }
  errors stderr
  tls off
}
:3000 {
  proxy / grafana:3000 {
    transparent
  }
  errors stderr
  tls off
}
```

A closer look to compose file: db

```
db:
  image: postgres:11-alpine
  restart: always
  environment:
    POSTGRES_PASSWORD: homerSeven
    POSTGRES_USER: root
  expose:
    - 5432
  restart: unless-stopped
  volumes:
    - ./conf/init-user-db.sh:/docker-entrypoint-initdb.d/init-user-db.sh
    - postgres_data:/var/lib/postgresql/data
  healthcheck:
    test: ["CMD-SHELL", "psql -h 'localhost' -U 'root' -c '\\1'"]
    interval: 1s
    timeout: 3s
    retries: 30
  networks:
    homer.server:
```

A closer look to compose file: heplify-server

```
heplify-server:
  image: sipcapture/heplify-server:master
  container_name: heplify-server
  ports:
    - "9060:9060"
    - "9060:9060/udp"
  command:
    - './heplify-server'
  environment:
    - "HEPLIFYSERVER HEPADDR=0.0.0.0:9060"
    - "HEPLIFYSERVER DBDRIVER=postgres"
    - "HEPLIFYSERVER DBHEMA=homer7"
    - "HEPLIFYSERVER DBADDR=db:5432"
    - "HEPLIFYSERVER DBUSER=root"
    - "HEPLIFYSERVER DBPASS=homerSeven"
    - "HEPLIFYSERVER DBDATATABLE=homer_data"
    - "HEPLIFYSERVER LOGLVL=info"
    - "HEPLIFYSERVER LOGSTD=true"
    - "HEPLIFYSERVER PROMADDR=0.0.0.0:9096"
    - "HEPLIFYSERVER PROMTARGETIP="
    - "HEPLIFYSERVER PROMTARGETNAME="
    - "HEPLIFYSERVER DBDROPDAYS=7"
    - "HEPLIFYSERVER DEDUP=false"
    - "HEPLIFYSERVER LOKIURL=http://loki:3100/api/prom/push"
    - "HEPLIFYSERVER LOKITIMER=2"
  restart: unless-stopped
  expose:
    - 9060
  depends_on:
    db:
      condition: service_healthy
```

A closer look to compose file: webapp

```
homer-webapp :  
  container name : homer-webapp  
  image : sipcapture/homer-app:master  
  environment :  
    - "DB_HOST=db"  
    - "DB_USER=root"  
    - "DB_PASS=homerSeven"  
  expose :  
    - 80  
  restart : unless-stopped  
  depends_on :  
    db:  
      condition : service_healthy
```

A closer look to compose file: TICK

```
telegraf:
  image: telegraf:alpine
  restart: unless-stopped
  volumes:
    - ./conf/telegraf.conf:/etc/telegraf/telegraf.conf
  ports:
    - "8092:8092/udp"
    - "8125:8125/udp"
  depends_on:
    - heplify-server
  networks:
    homer.server:
```

```
influxdb:
  image: influxdb:alpine
  volumes:
    - influx_data:/var/lib/influxdb
  expose:
    - 8086
  networks:
    homer.server:
```

A closer look to compose file: TICK

```
chronograf:
  image: chronograf:alpine
  environment:
    INFLUXDB_URL: http://influxdb:8086
    KAPACITOR_URL: http://kapacitor:9092
  expose:
    - 8888
  links:
    - influxdb
    - kapacitor
  networks:
    homer.server:

kapacitor:
  image: kapacitor:alpine
  environment:
    KAPACITOR_HOSTNAME: kapacitor
    KAPACITOR_INFLUXDB_0_URLS_0: http://influxdb:8086
  links:
    - influxdb
  expose:
    - 9092
  networks:
    homer.server:
```

A closer look to compose file: grafana

```
grafana:
  image: grafana/grafana:master
  links:
    - influxdb
  volumes:
    - grafana data:/var/lib/grafana
    - ./grafana/provisioning:/etc/grafana/provisioning/
  environment:
    - GF_SECURITY_ADMIN_USER=${ADMIN_USER:-admin}
    - GF_SECURITY_ADMIN_PASSWORD=${ADMIN_PASSWORD:-sipcapture}
    - GF_USERS_ALLOW_SIGN_UP=false
    - GF_EXPLORE_ENABLED=true
  restart: unless-stopped
  expose:
    - 3000
  networks:
    homer.server:
```


A closer look to compose file: loki

```
loki:
  image: grafana/loki:master
  container_name: loki
  volumes:
    - ./loki/loki-local-config.yaml:/etc/loki/loki-local-config.yaml
    - loki data:/data/loki
  restart: unless-stopped
  expose:
    - 3100
  ports:
    - "3100:3100"
  command: "-config.file=/etc/loki/loki-local-config.yaml"
  labels:
    org.label-schema.group: "monitoring"
  networks:
    homer.server:
```

A closer look to compose file: provisioning

command:

...

```
curl -G http://influxdb:8086/query --data-urlencode 'q=CREATE DATABASE homer'
curl -G http://influxdb:8086/query --data-urlencode 'q=CREATE RETENTION POLICY "60s" ON
"homer" DURATION 4w REPLICATION 1 DEFAULT'
curl -G http://influxdb:8086/query --data-urlencode 'q=CREATE RETENTION POLICY "300s" ON
"homer" DURATION 8w REPLICATION 1'
curl -G http://influxdb:8086/query --data-urlencode 'q=CREATE RETENTION POLICY "3600s"
ON "homer" DURATION 16w REPLICATION 1'
curl -G http://influxdb:8086/query --data-urlencode 'q=CREATE RETENTION POLICY "86400s"
ON "homer" DURATION 32w REPLICATION 1'
curl -G http://influxdb:8086/query --data-urlencode 'q=CREATE CONTINUOUS QUERY cq_300s
ON homer RESAMPLE EVERY 1m BEGIN SELECT mean(*) INTO homer."300s".:MEASUREMENT FROM
homer."60s"../*/ GROUP BY time(5m),* END'
curl -G http://influxdb:8086/query --data-urlencode 'q=CREATE CONTINUOUS QUERY cq_3600s
ON homer RESAMPLE EVERY 10m BEGIN SELECT mean(*) INTO homer."3600s".:MEASUREMENT FROM
homer."60s"../*/ GROUP BY time(1h),* END'
curl -G http://influxdb:8086/query --data-urlencode 'q=CREATE CONTINUOUS QUERY cq_86400s
ON homer RESAMPLE EVERY 1h BEGIN SELECT mean(*) INTO homer."86400s".:MEASUREMENT FROM
homer."60s"../*/ GROUP BY time(1d),* END'
curl -G http://influxdb:8086/query --data-urlencode 'q=drop retention policy "autogen"
ON "homer"'
```

...

A closer look to compose file: provisioning

command:

...

```
echo Chronograf Dashboard push ...;
```

```
curl -i -X POST -H "Content-Type: application/json"
```

```
http://chronograf:8888/chronograf/v1/dashboards -d @/tmp/homer_dashboard.json
```

```
echo Provisioning completed! Exiting ...;
```

...

Testing your H7: hepgen.js

Barebone HEP Packet Generator <https://github.com/sipcapture/hepgen.js>

```
var config = {
  HEP_SERVER: '127.0.0.1',
  HEP_PORT: 9060,
  HEP_ID: '2001',
  HEP_AUTH: 'myHep',
  MESSAGES: [
    {
      rcinfo: { <HEP_HEADER> },
      pause: 0,
      payload: <sip message>
    },
    {
      rcinfo: { <HEP_HEADER> },
      pause: 1000,
      payload: <sip message>
    }
  ]
}
```

Testing your H7: hepgen.js

Requirements:

- nodejs 10+ <https://github.com/nodesource/distributions/blob/master/>

```
git clone https://github.com/sipcapture/hepgen.js
cd hepgen.js
npm install
node hepgen.js -c ./config/b2bcall_rtcp_log.js
```

Testing your H7: hepgen.js

Requirements:

- docker-ce

<https://docs.docker.com/install/>

```
docker pull sipcapture/hepgenjs:master
docker run --rm --network=hom7-influx-graf-loki_homer.server sipcapture/hepgenjs:master
/bin/bash

node hepgen.js -c ./config/b2bcall_rtcp_log.js
```