Functional programming @ Avast

Jenda Kolena

Scala dev, Avast Threat Labs

About Avast

Cyber-security software company, started and still headquartered in Prague.

One of the biggest AV companies in the world.

avast

Many teams in Avast do FP, mostly in Scala.

More about Scala

Hybrid of Java, Haskell and ML - fusion of OOP and FP.

Powerful type system - comparable to Haskell.

Runs on JVM - interoperability with Java, Kotlin, Groovy, ...

More about Scala

Type-classes.

Immutable data classes - case classes.

Pattern matching.

For-comprehension == do-notation.

Avast Scala

Purely functional programming (we try).

Monix library - asynchronous programming, separation of side-effects.

Cats library - Functor, Monad, etc., advanced syntax

avast

We are the biggest Scala employer in Prague.

Probably even in whole Czechia.

Threat Labs

We are the core department of antivirus company.

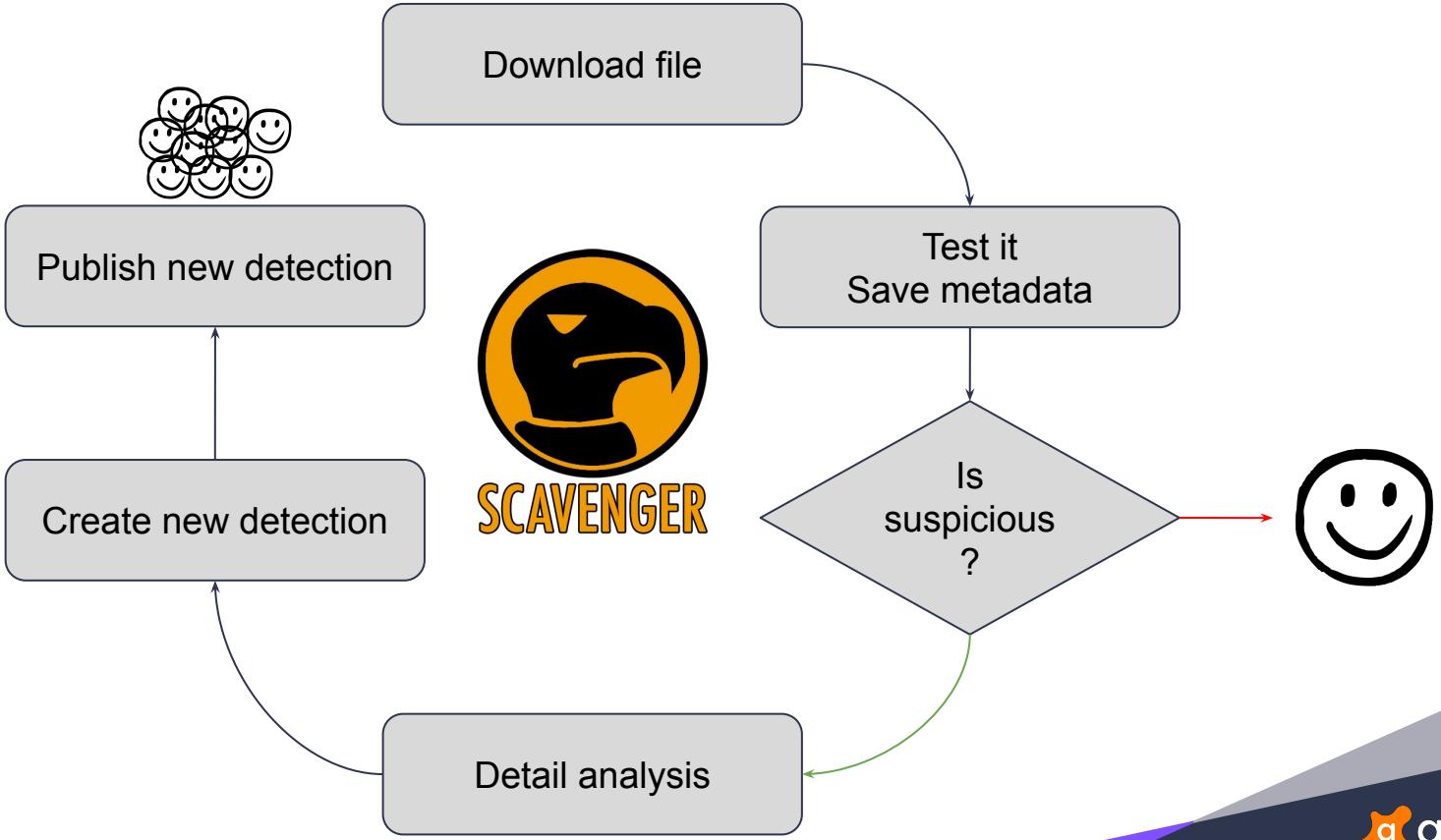
We do malware research and develop supporting systems.

avast

SCAVENGER

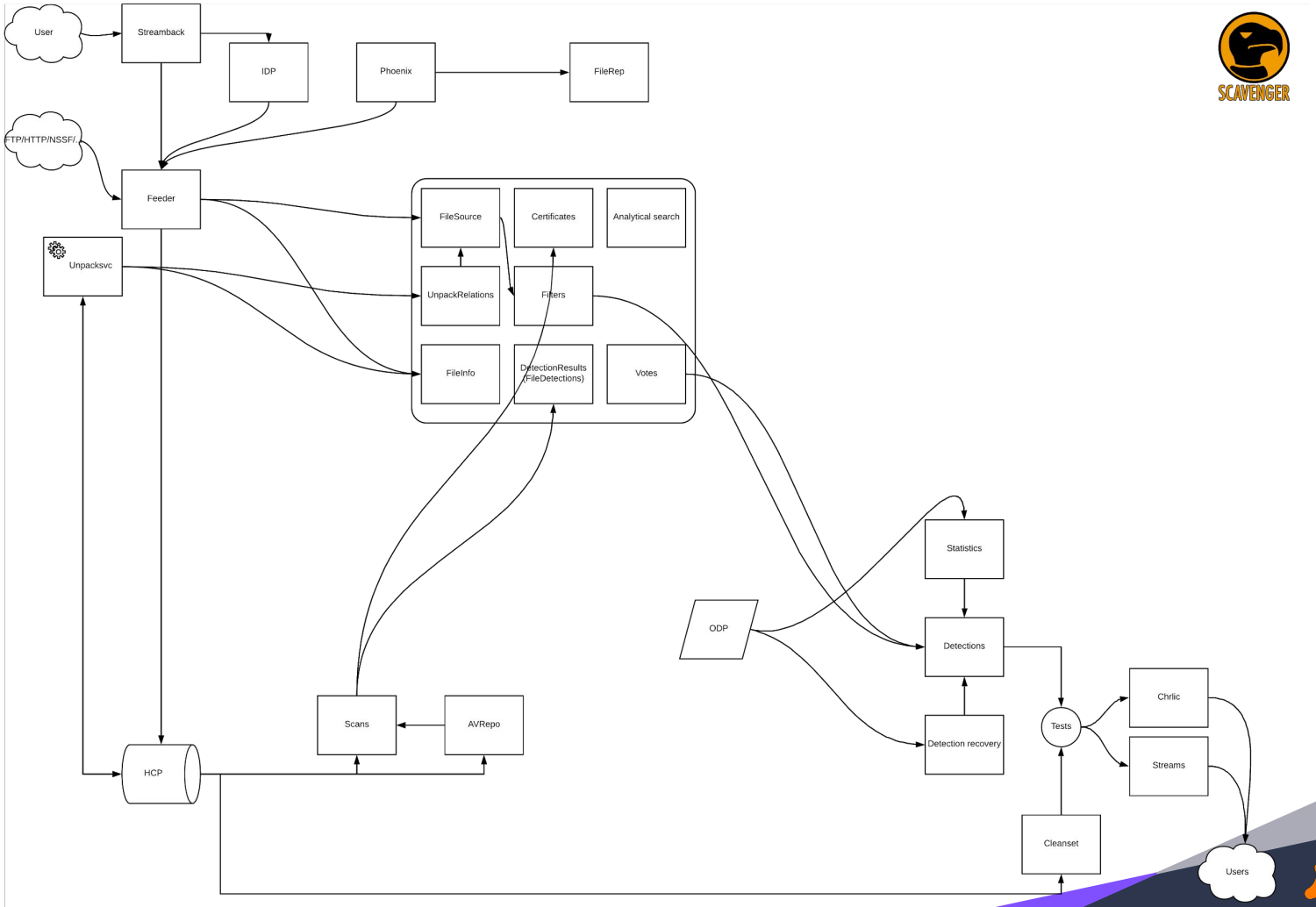
avast

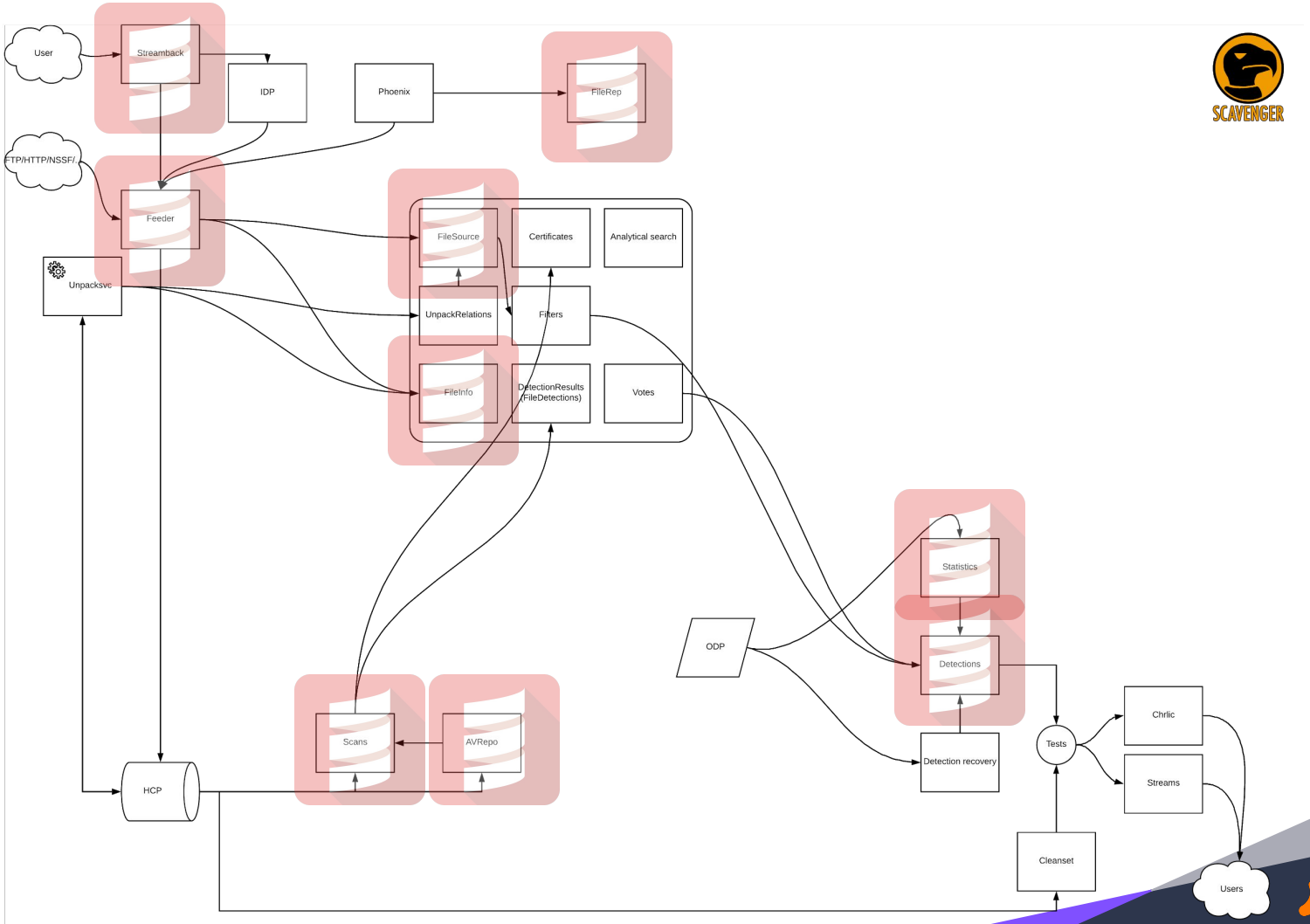
Threat Labs is a virus laboratory.



We run a number of reactive microservices.

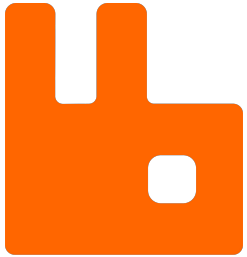
Written in Scala. And Python. And Perl.


User
Streamback
IDP
Phoenix
FileRep
FTP/HTTP/NSSF/
Feeder
Unpacksvc
FileSource
Certificates
Analytical search
UnpackRelations
Filters
FileInfo
DetectionResults (FileDetections)
Votes
Statistics
ODP
Detections
Detection recovery
Tests
Chrlic
Streams
Scans
AVRepo
HCP
Cleanset
Users


User
Streamback
IDP
Phoenix
FileRep
FTP/HTTP/NSSF/
Feeder
Unpacksvc
FileSource
Certificates
Analytical search
UnpackRelations
Filters
FileInfo
DetectionResults
(FileDetections)
Votes
Statistics
ODP
Detections
Scans
AVRepo
Detection recovery
Tests
Chrlic
Streams
HCP
Cleanset
Users

avast

We use gRPC and RabbitMQ for inter-service communication.



RabbitMQ

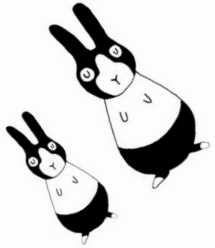
```scala
object RabbitMqConfiguration extends TaskLogging {
  def apply(rabbitEc: ExecutorService, config: Config, monitor: Monitor): Task[Unit] = {

    val rabbitConnection: Task[DefaultRabbitMQConnection[Task]] = {
      RabbitMQConnection.fromConfig[Task](config.getConfig( path = "rabbit"), rabbitEc)
    }

    def consumer(rabbitConnection: RabbitMQConnection[Task]): Task[RabbitMQConsumer[Task]] = {
      rabbitConnection
        .newConsumer( configName = "fileDataConsumer", monitor.named( name = "consumer")) {
          case Delivery.Ok(body, properties, routingKey) =>
            logger.debug(s"Received: $body, $properties, $routingKey") >>
              Task.pure(DeliveryResult.Ack)

          case Delivery.MalformedContent(_, _, _, exception) =>
            logger.warn( message = "Error while receiving event", exception) >>
              Task.pure(DeliveryResult.Reject)
        }
    }

    for {
      conn <- rabbitConnection
      _ <- consumer(conn)
    } yield ()
  }
}
```



gRPC

```
service ExampleService {
    rpc AddFileData(AddFileDataParams) returns (google.protobuf.Empty) {
        option (roles) = "Admin";
        option (roles) = "Analyst";
    }
    rpc GetFileData(GetFileDataParams) returns (GetFileDataResult) {}
}

message AddFileDataParams {
    bytes file_id = 1;
    string data = 2;
}

message GetFileDataParams {
    bytes file_id = 1;
}

message GetFileDataResult {
    repeated FileData file_data = 1;
}

message FileData {
    bytes file_id = 1;
    string data = 2;
}
```

avast

gRPC

```
trait ExampleServiceClient[F[_]] extends GrpcClient {
  def addFileData(fs: FileData): F[ServerResponse[Unit]]

  def getFileData(f: Sha256): F[ServerResponse[Seq[FileData]]]
}
```

avast

avast

We run services in Kubernetes cluster.

kubernetes

LUFT

Declarative deployment.

Infrastructure as a code.

avast

Q&A