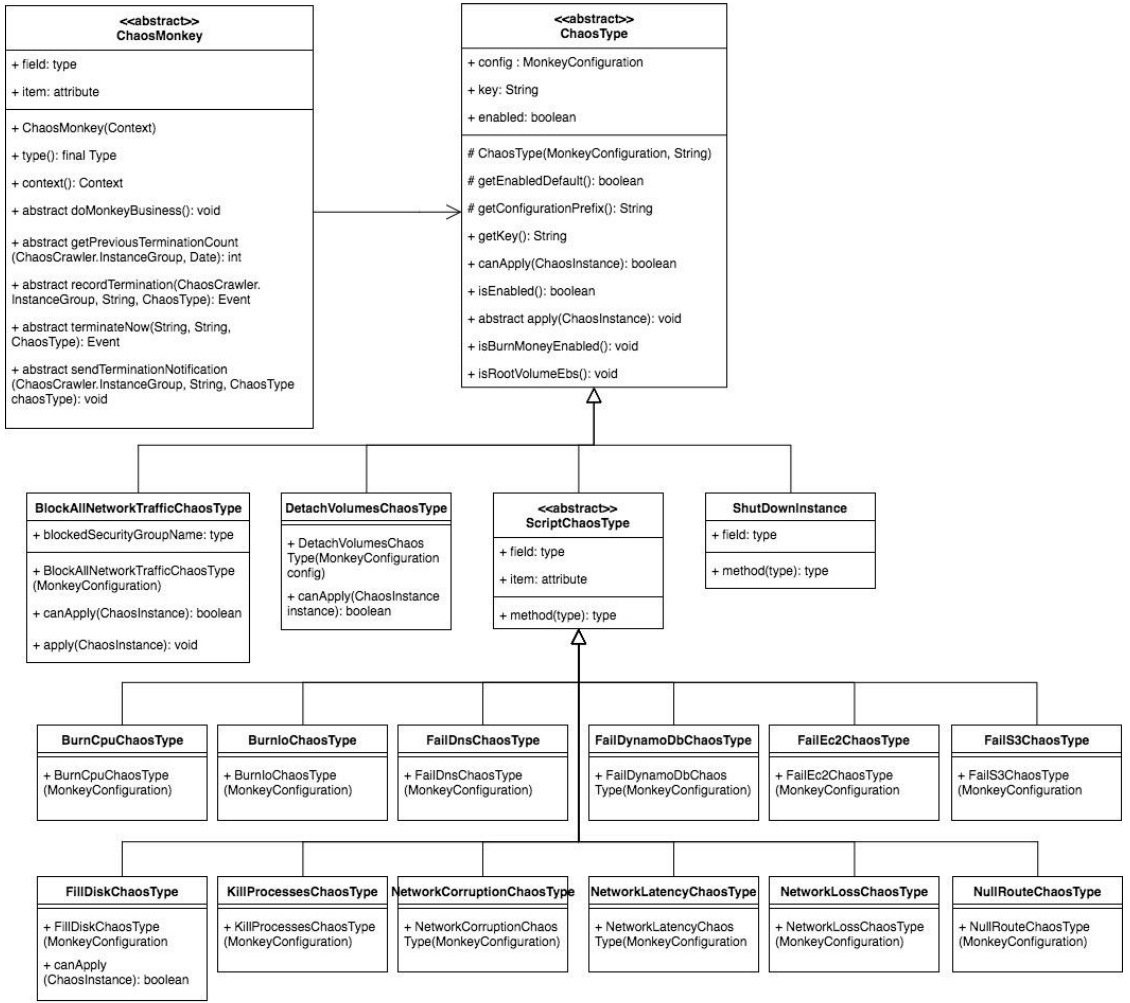
# Simian Army

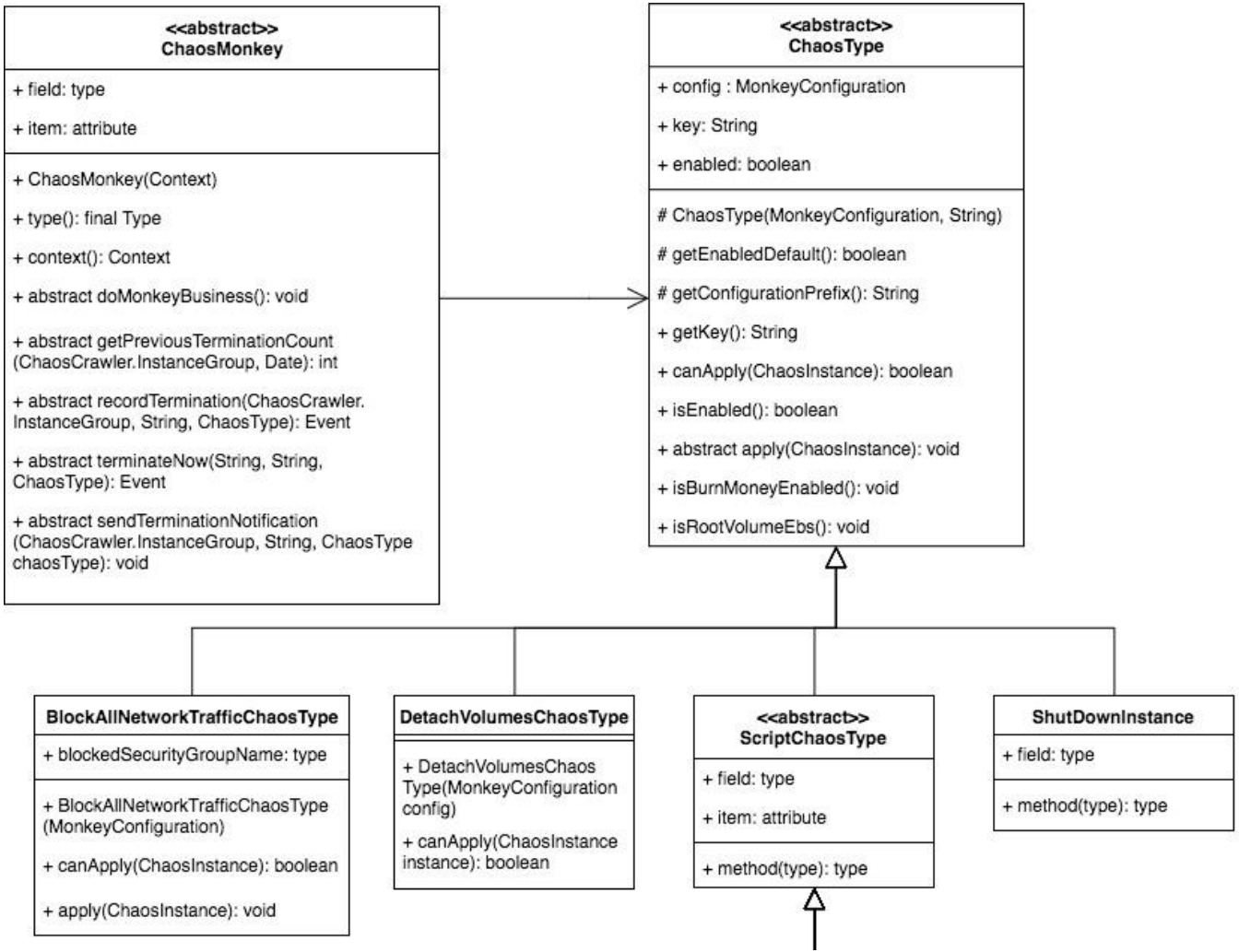
Chaos Monkey

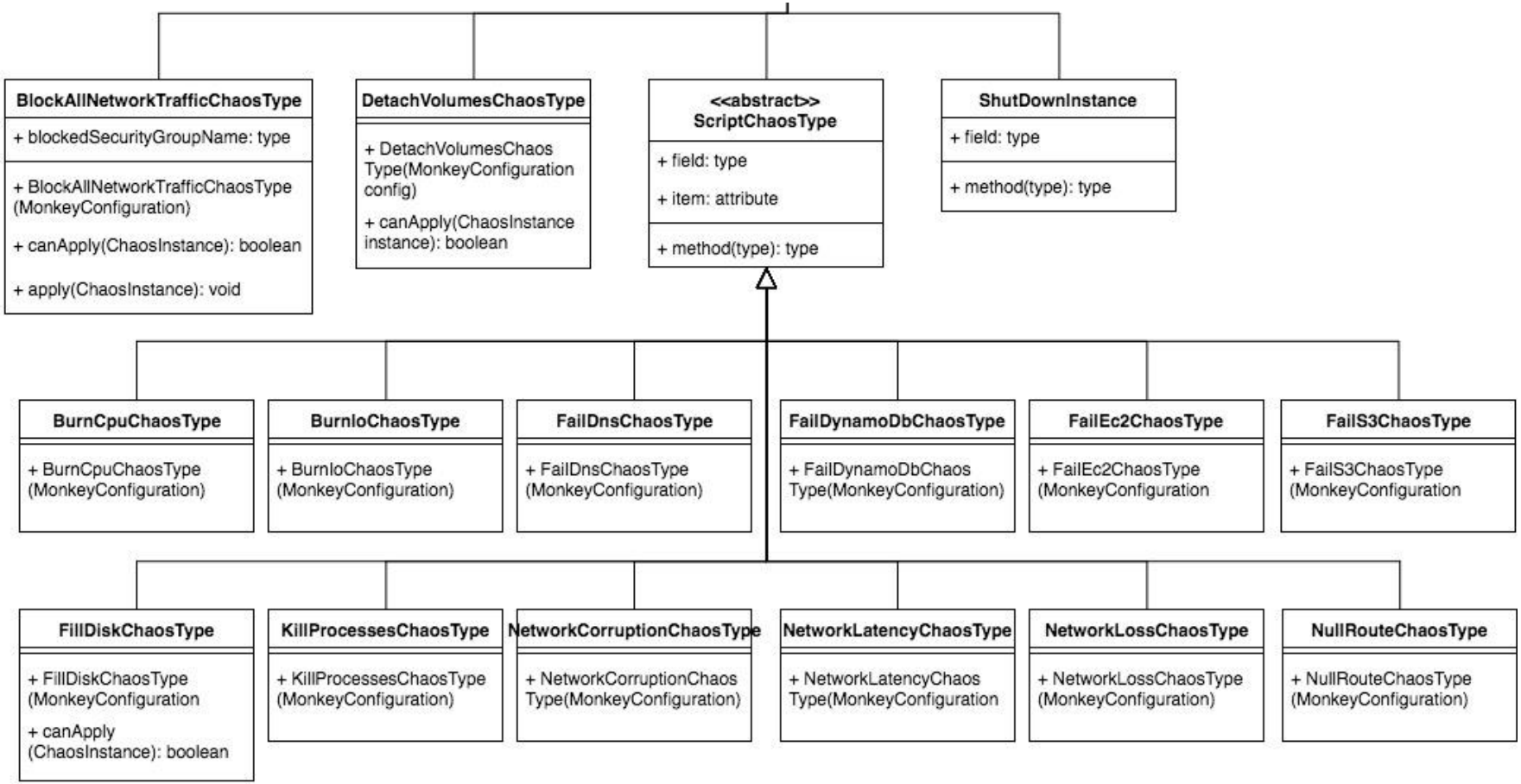
# Introduction

- What is Chaos Monkey
  - identifies groups of systems
  - randomly terminates one of the systems in a group
- Why run Chaos Monkey
  - Failures happen
  - Even if your architecture can tolerate a system failure,
    are you sure it will still be able to next week, how about next month?
  - Do your traffic load balancers correctly detect and route requests
    around system failures?

# Class diagram




<<abstract>>
ChaosMonkey
+ field: type
+ item: attribute
+ ChaosMonkey(Context)
+ type(): final Type
+ context(): Context
+ abstract doMonkeyBusiness(): void
+ abstract getPreviousTerminationCount (ChaosCrawler.InstanceGroup, Date): int
+ abstract recordTermination(ChaosCrawler. InstanceGroup, String, ChaosType): Event
+ abstract terminateNow(String, String, ChaosType): Event
+ abstract sendTerminationNotification (ChaosCrawler.InstanceGroup, String, ChaosType chaosType): void
<<abstract>>
ChaosType
+ config : MonkeyConfiguration
+ key: String
+ enabled: boolean
# ChaosType(MonkeyConfiguration, String)
# getEnabledDefault(): boolean
# getConfigurationPrefix(): String
+ getKey(): String
+ canApply(ChaosInstance): boolean
+ isEnabled(): boolean
+ abstract apply(ChaosInstance): void
+ isBurnMoneyEnabled(): void
+ isRootVolumeEbs(): void
BlockAllNetworkTrafficChaosType
+ blockedSecurityGroupName: type
+ BlockAllNetworkTrafficChaosType (MonkeyConfiguration)
+ canApply(ChaosInstance): boolean
+ apply(ChaosInstance): void
DetachVolumesChaosType
+ DetachVolumesChaos Type(MonkeyConfiguration config)
+ canApply(ChaosInstance instance): boolean
<<abstract>>
ScriptChaosType
+ field: type
+ item: attribute
+ method(type): type
ShutDownInstance
+ field: type
+ method(type): type


BlockAllNetworkTrafficChaosType
+ blockedSecurityGroupName: type
+ BlockAllNetworkTrafficChaosType (MonkeyConfiguration)
+ canApply(ChaosInstance): boolean
+ apply(ChaosInstance): void
DetachVolumesChaosType
+ DetachVolumesChaos Type(MonkeyConfiguration config)
+ canApply(ChaosInstance instance): boolean
<<abstract>> ScriptChaosType
+ field: type
+ item: attribute
+ method(type): type
ShutDownInstance
+ field: type
+ method(type): type
BurnCpuChaosType
+ BurnCpuChaosType (MonkeyConfiguration)
BurnIoChaosType
+ BurnIoChaosType (MonkeyConfiguration)
FailDnsChaosType
+ FailDnsChaosType (MonkeyConfiguration)
FailDynamoDbChaosType
+ FailDynamoDbChaos Type(MonkeyConfiguration)
FailEc2ChaosType
+ FailEc2ChaosType (MonkeyConfiguration
FailS3ChaosType
+ FailS3ChaosType (MonkeyConfiguration
FillDiskChaosType
+ FillDiskChaosType (MonkeyConfiguration
+ canApply (ChaosInstance): boolean
KillProcessesChaosType
+ KillProcessesChaosType (MonkeyConfiguration)
NetworkCorruptionChaosType
+ NetworkCorruptionChaos Type(MonkeyConfiguration)
NetworkLatencyChaosType
+ NetworkLatencyChaos Type(MonkeyConfiguration
NetworkLossChaosType
+ NetworkLossChaosType (MonkeyConfiguration)
NullRouteChaosType
+ NullRouteChaosType (MonkeyConfiguration)

# Functionality

| | |
|---|---|
| BurnCpuChaosType | 在node上跑高CPU的程式，盡可能用到所有 CPU |
| BurnIoChaosType | 在node上跑高I/O的程式 |
| FailDnsChaosType | 擋住TCP UDP的port 53，讓DNS壞掉 |
| FailDynamoDbChaosType | 在 /etc/hosts路徑加東西，讓Dynamo DB的API壞掉 |
| FailEc2ChaosType | 在 /etc/hosts路徑加東西，讓EC2的API壞掉 |
| FailS3ChaosType | 在 /etc/hosts路徑加東西，讓S3的API壞掉 |
| FillDiskChaosType | 創造一個很大的檔案，讓硬碟爆滿 |
| KillProcessesChaosType | 在node上強迫某process的執行 |
| NullRouteChaosType | 將某個node弄offline<br>目前是offline 10.x.x.x的aws網路區段 |

# Design

- Design Pattern
  - Is Chaos Monkey using “Strategy Pattern” ?
- Design Principle
  - Encapsulate what varies