

D-SCRIPT: スクリプト技術による柔軟な モニタリングとディペンダビリティ コントロールの実現

倉光君郎, 横浜国立大学
JST/CREST DEOS Project

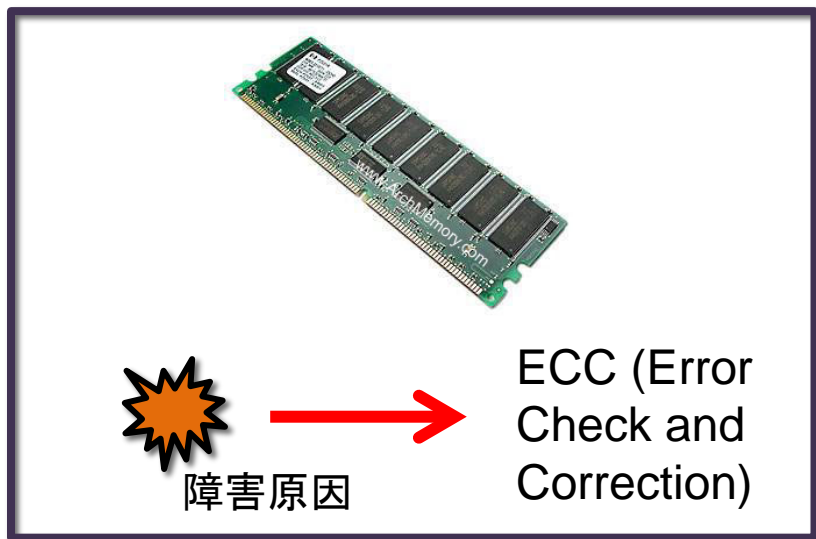


本日の内容

- D-Script とは何か？
- DEOS プロセス, D-Case, D-RE との関係
- D-Script の実行モデル
 - 既存のスクリプトと何が違うのか？
- D-Script の参照実装
 - KonohaScript <http://www.KonohaScript.org/>

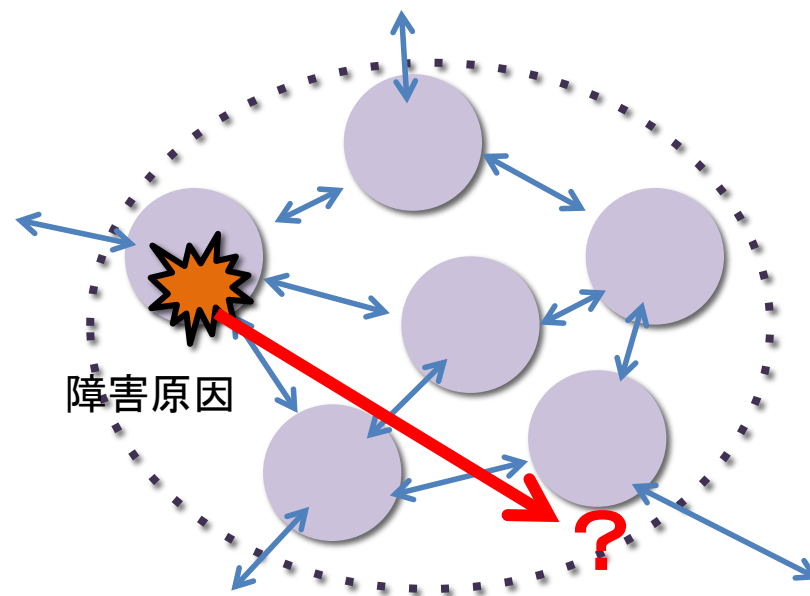
オープンシステム時代のディペンダビリティ対応

クローズド システム



障害原因とその対応の関係が明確
障害対応を作り込める

オープンシステム



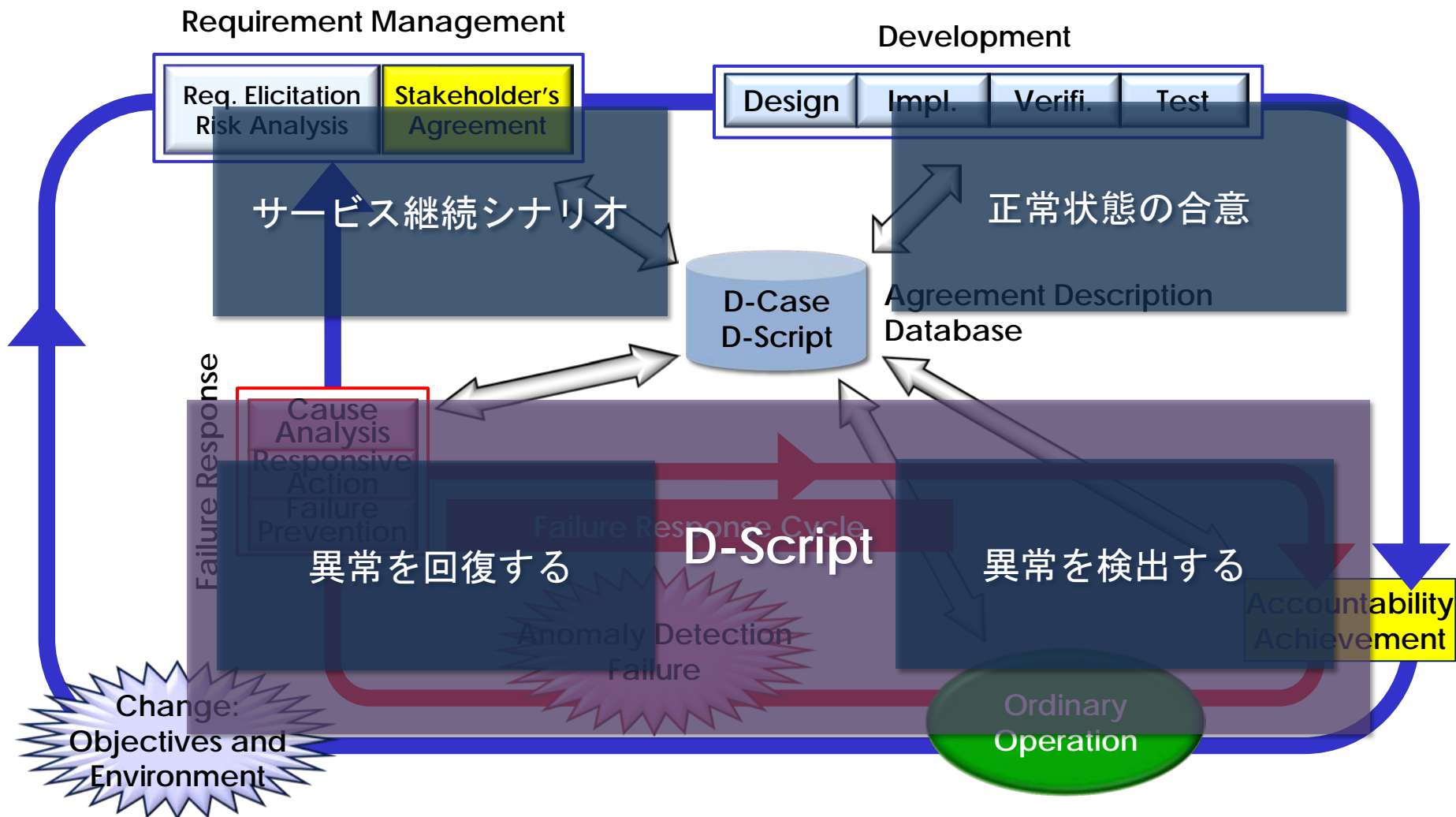
障害原因とその対応の関係が不明確
障害対応を作り込めない

D-Script : スクリプトによる外部記述

- キー・アイディア
 - スクリプト技術を、動的かつ柔軟な拡張性に用いて、
 - ディペンダビリティマネジメント（メンテナンスから障害処理）まで記述する
- 我々が提供するもの
 - 柔軟さと信頼性（堅牢性）の両立
 - スクリプト実行のプロセス化
 - アクセス制御（セキュリティ）
 - 障害対策の説明責任支援（実行ログの自動化）

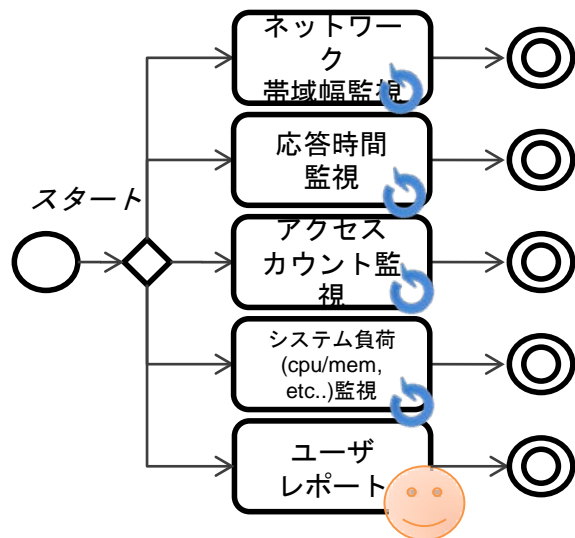
DEOS プロセスと D-Script

Change Accommodation Cycle

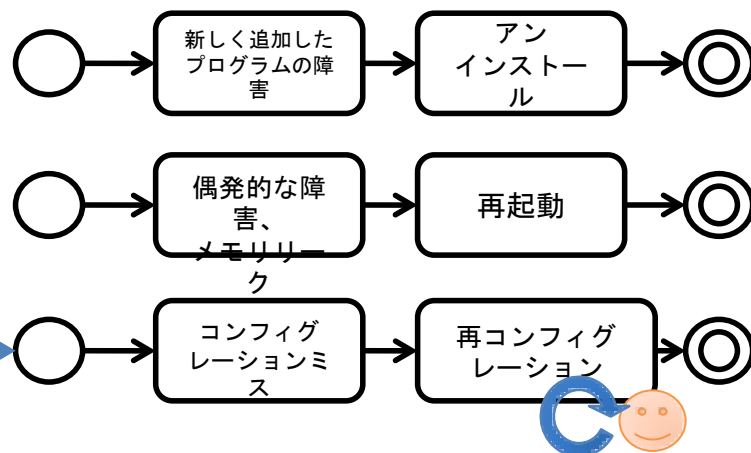


サービス継続シナリオ (BPMN表記)

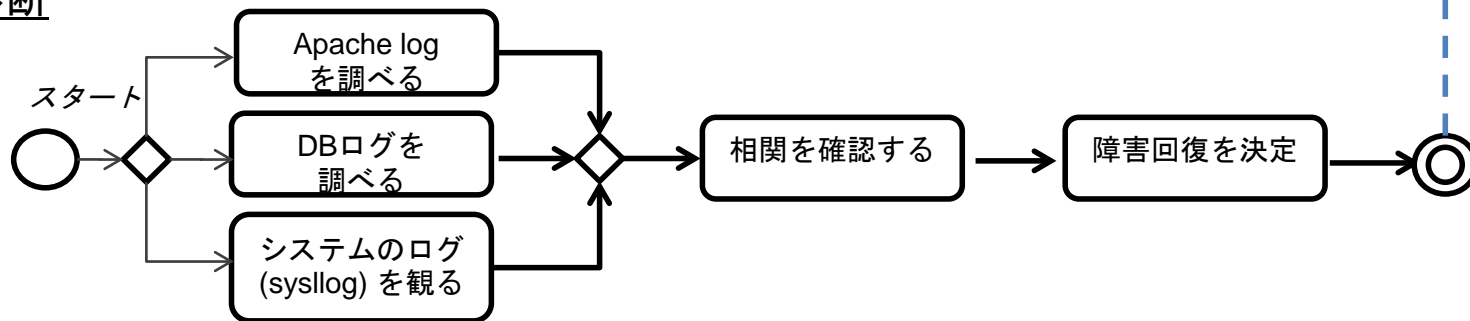
異常検出



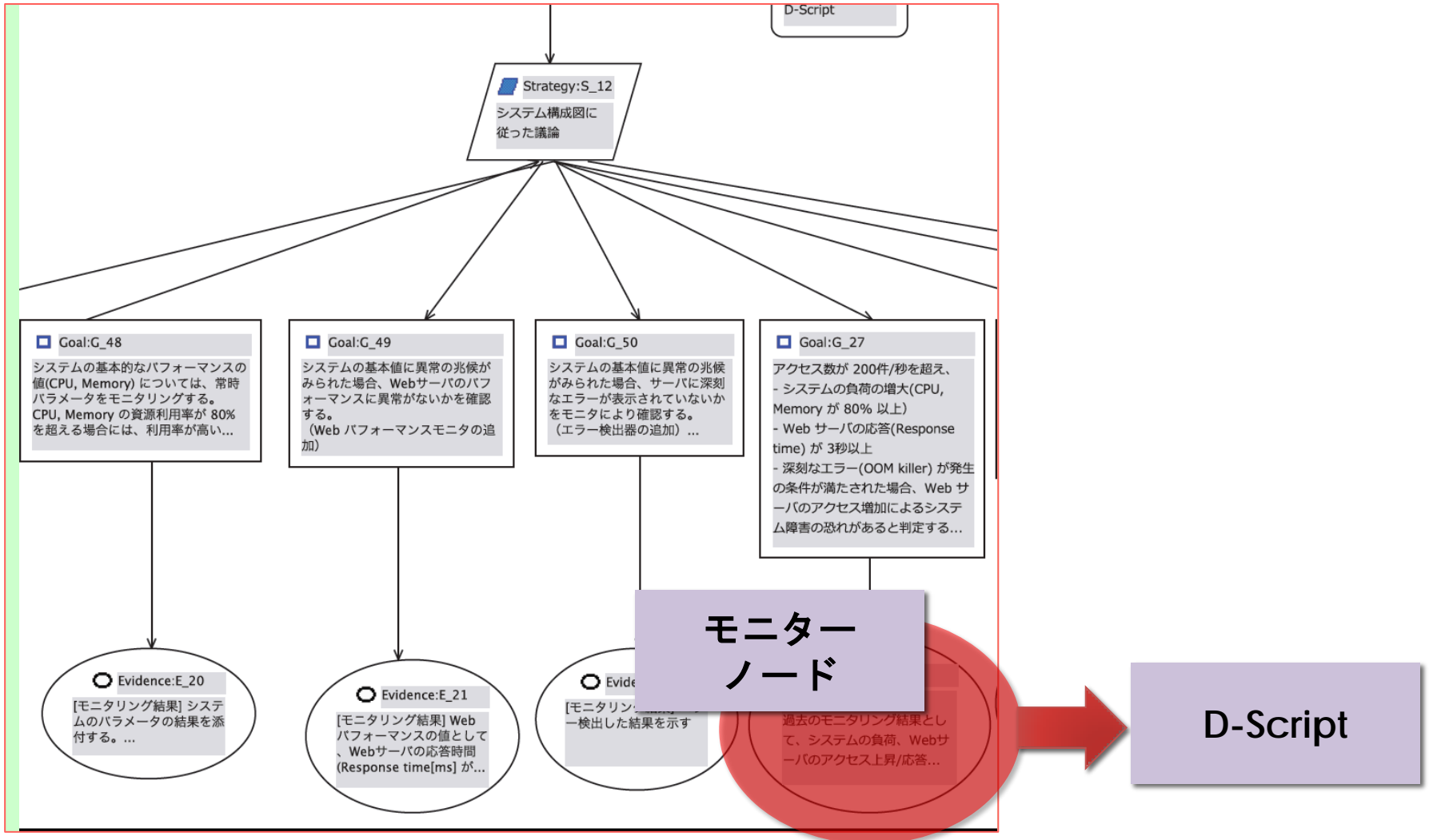
障害からの回復プラン



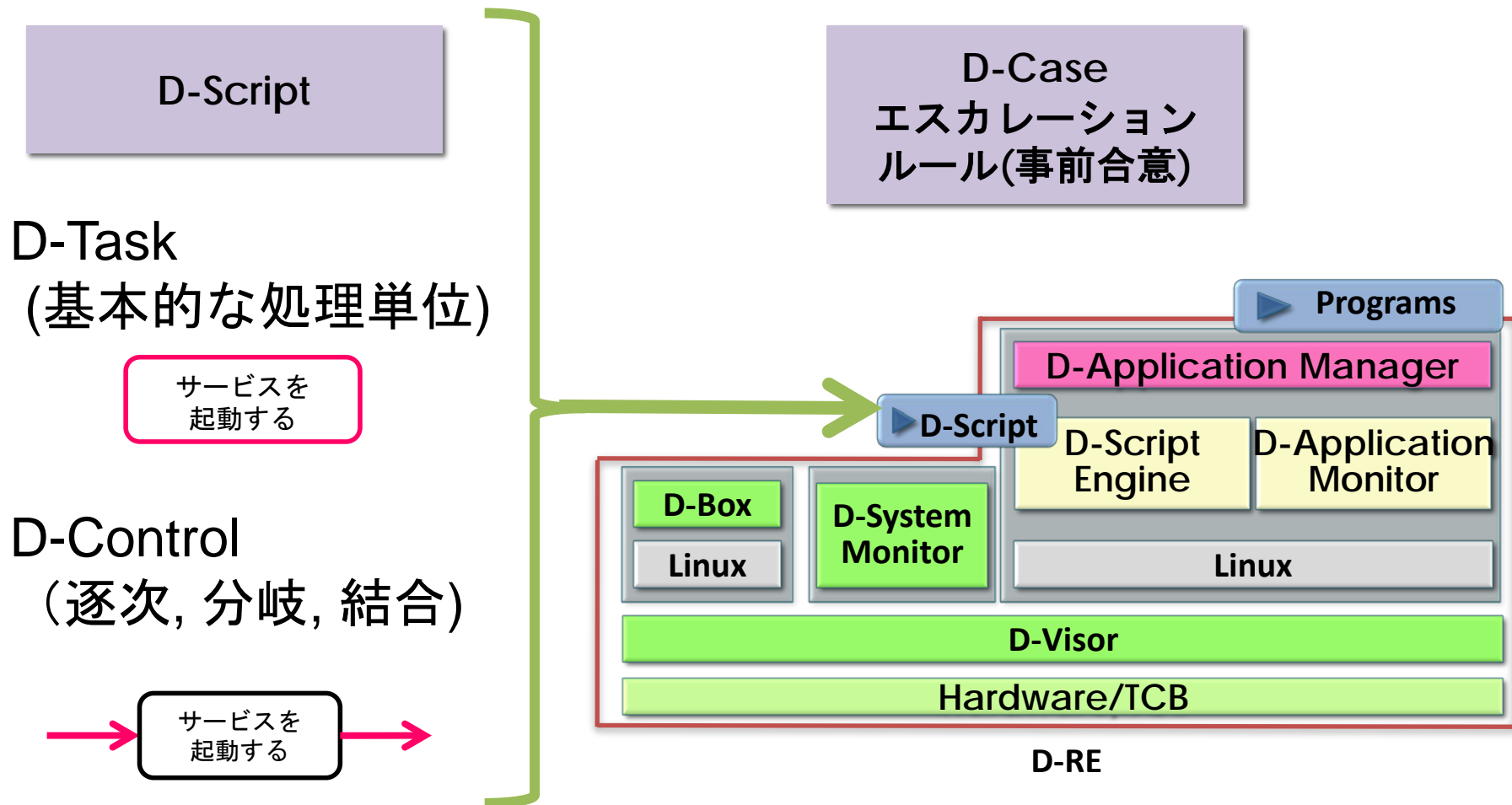
異常診断



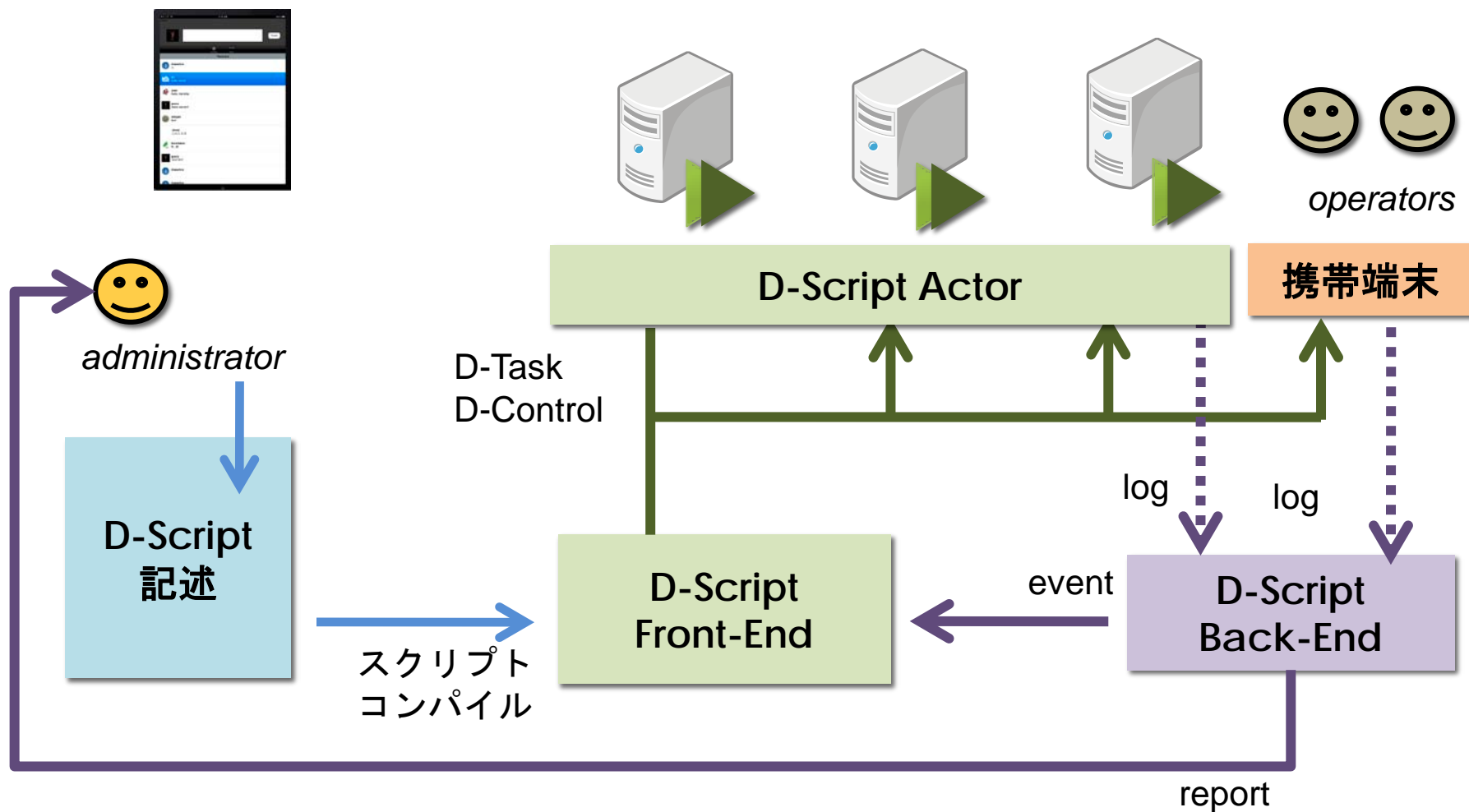
D-Case : 正常の合意とモニターノード



D-RE と D-Script (D-Task + D-Control)



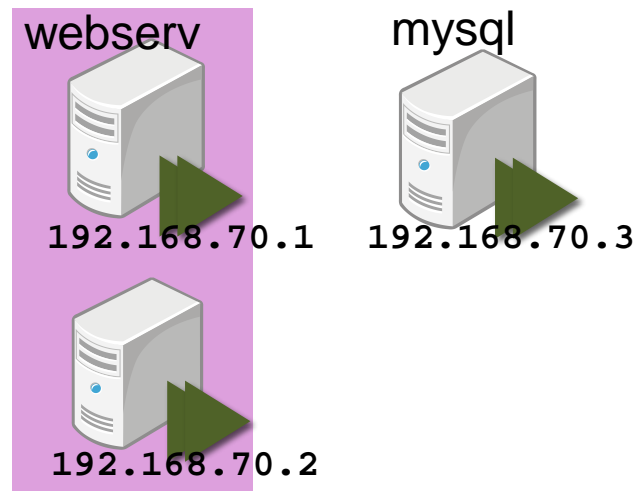
D-Script アーキテクチャ(分散系)



D-Script 言語 (Domain Specific Language)

D-Taskを定義する

```
taskset ApacheTask {  
  task StartWeb () {  
    os.init("/etc/..");  
  }  
  task StopWeb () {  
    os.kill(pid);  
  }  
}
```



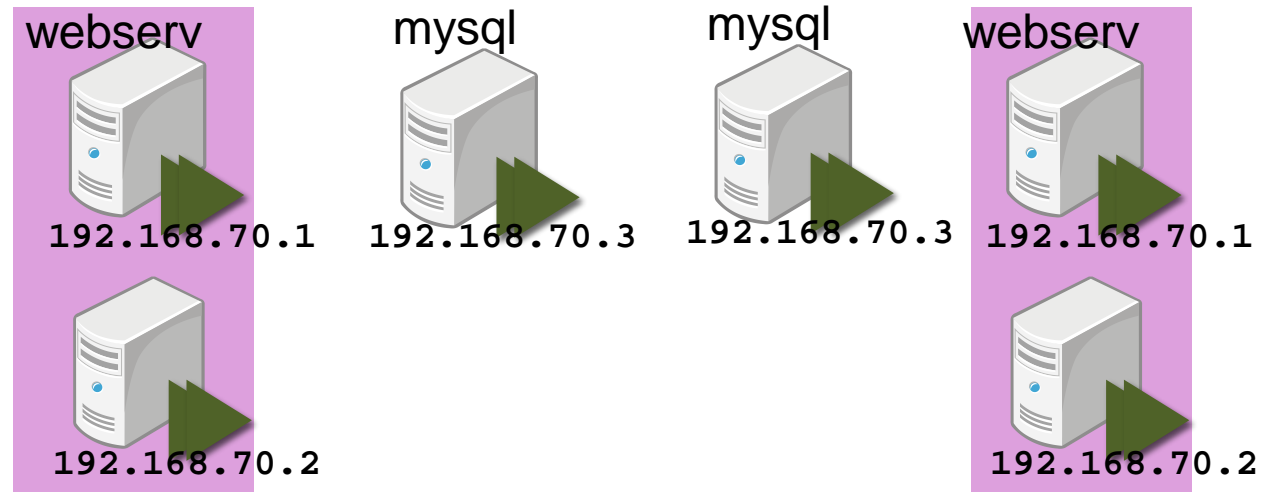
D-Script Actor とバインドする

```
bind web with ApacheTask = [  
  actor:192.168.70.1,  
  actor:192.168.70.2  
]
```

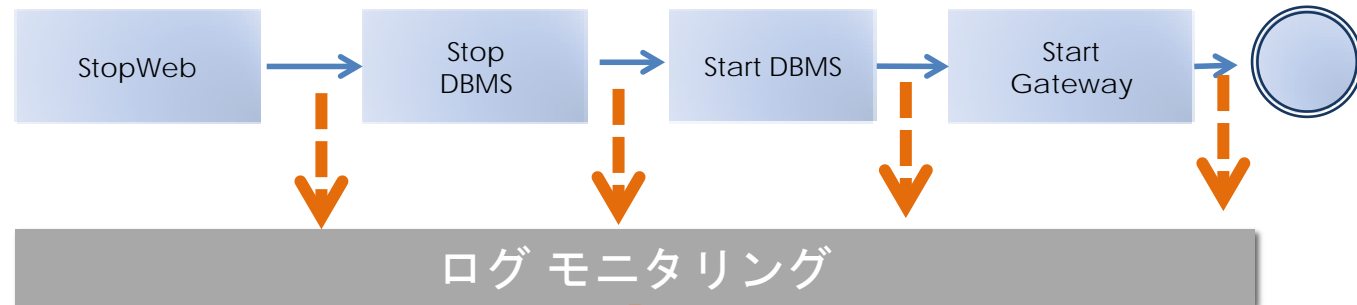
D-Task の実行を制御する

```
control {  
  web.StopWeb()  
  mysql.StopDB()  
  ...  
  web.StartWeb()  
}
```

D-Script の実行モデル (監視しながら実行する)

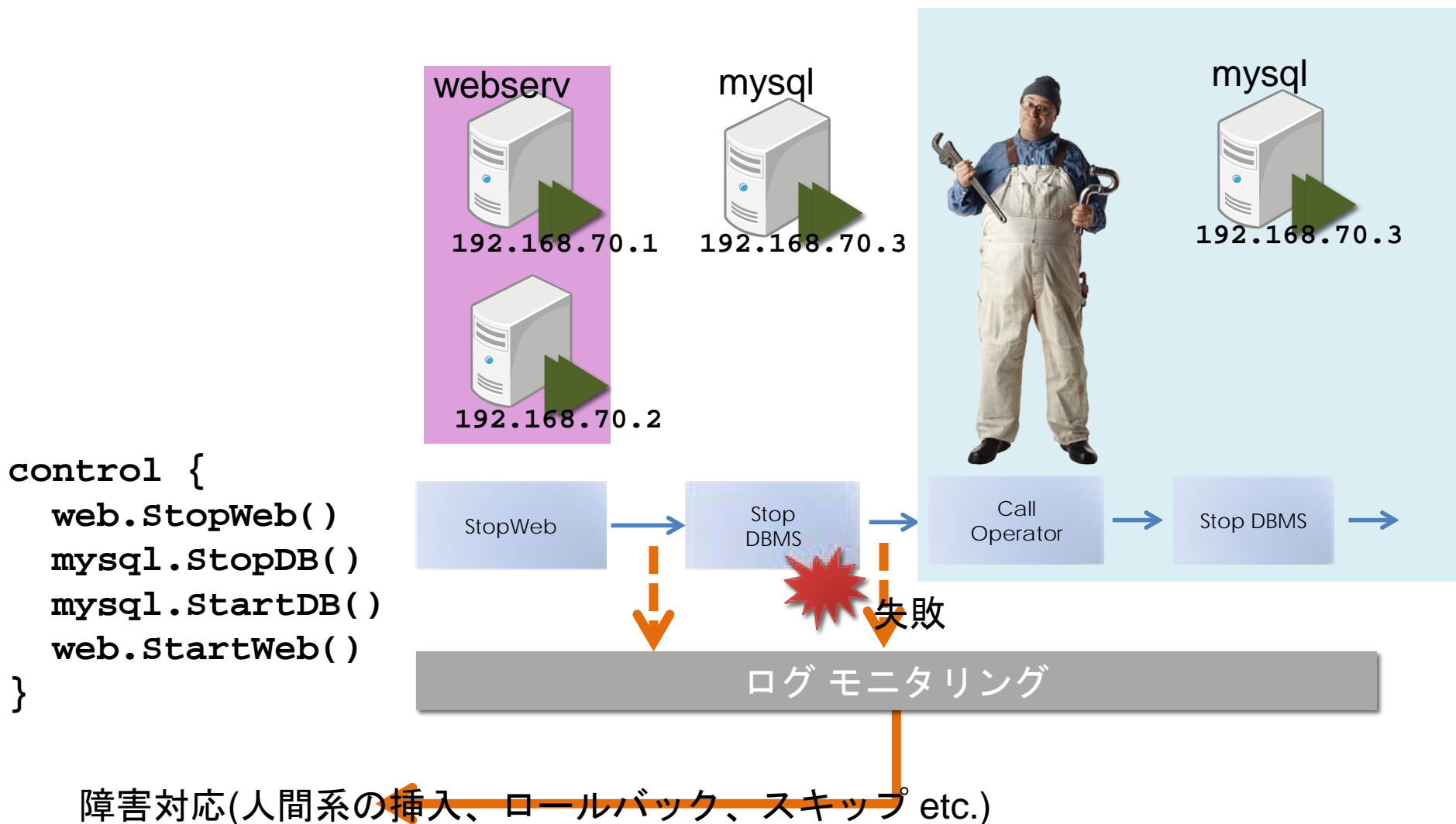


```
control {
  web.StopWeb()
  mysql.StopDB()
  mysql.StartDB()
  web.StartWeb()
}
```

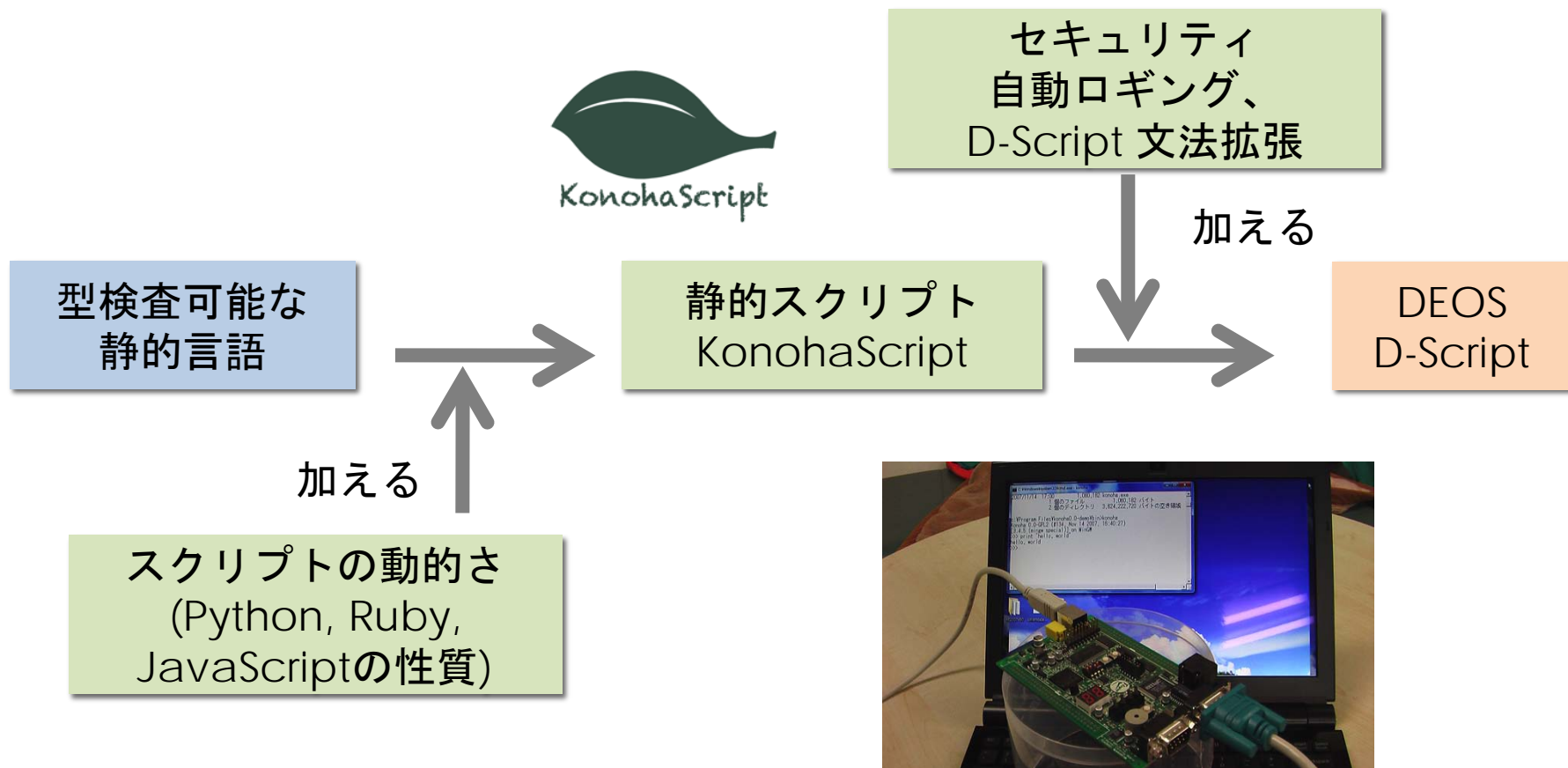


フィードバック

D-Script の実行モデル (部分障害への対応)



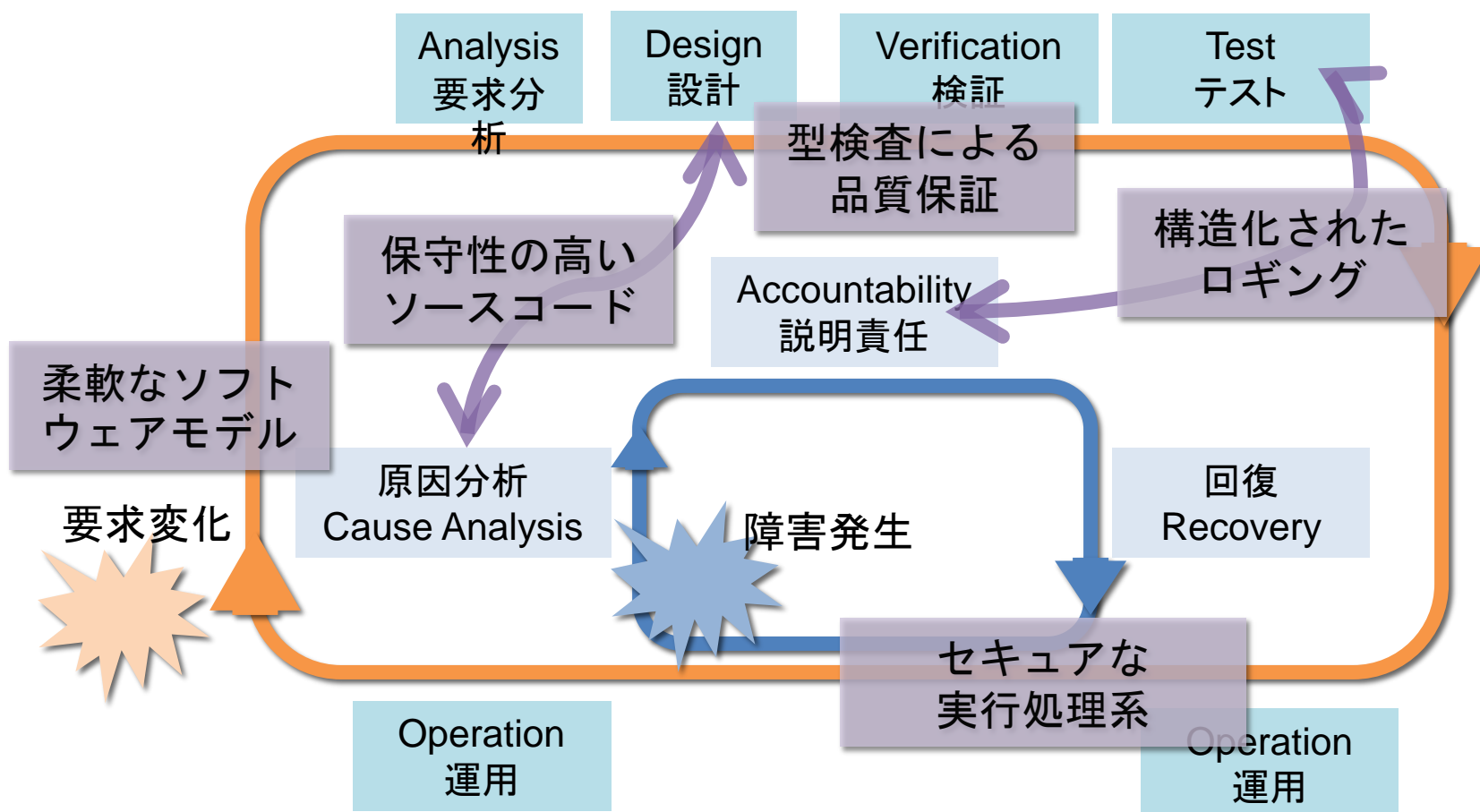
D-Script の参照実装



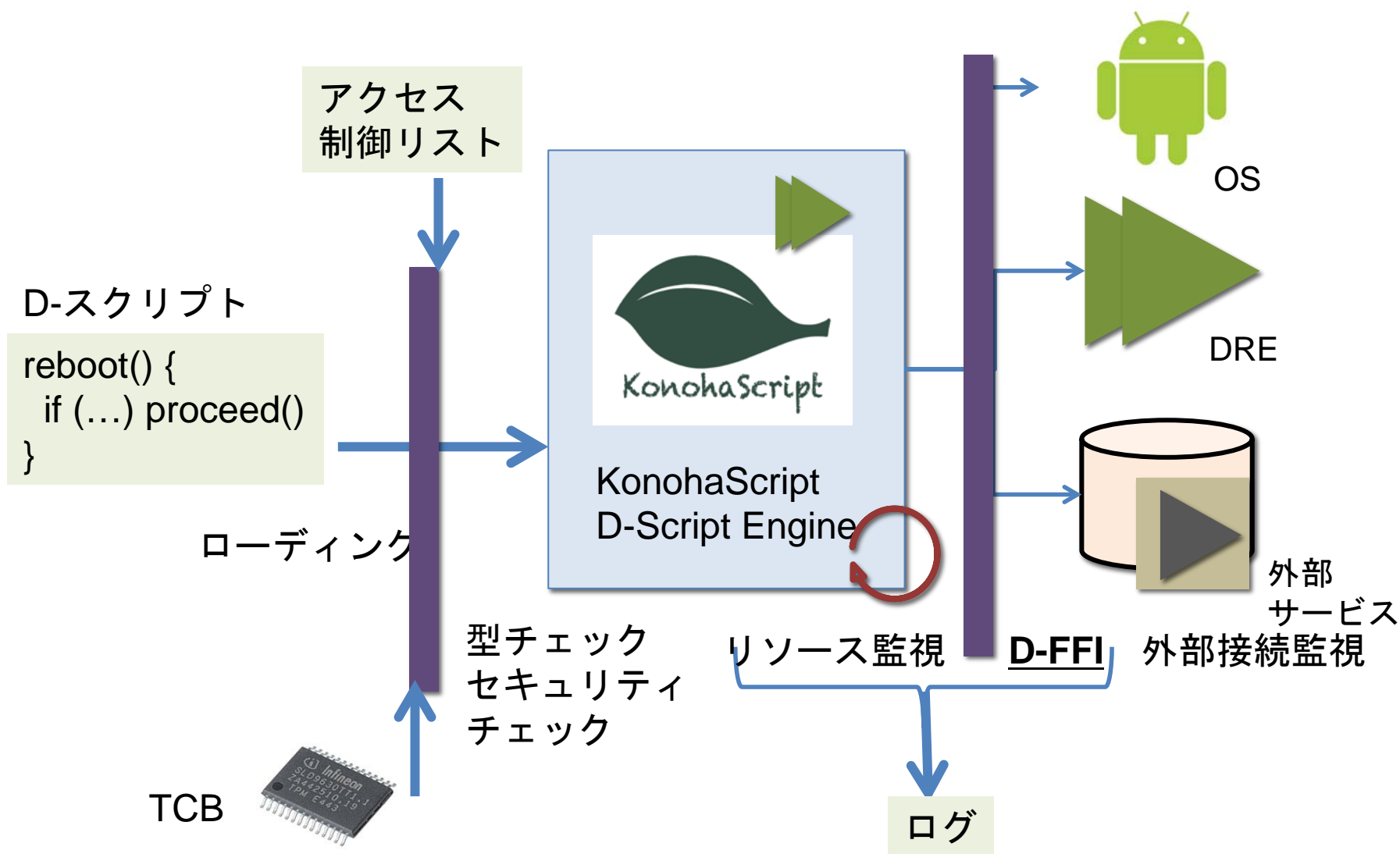
KonohaScript は、軽量な実装で組み込みシステム上でも動作可能

DEOS プロセスから D-Script の技術要求

- DEOSプロセスの各フェーズでオープンシステムの実現をサポートする

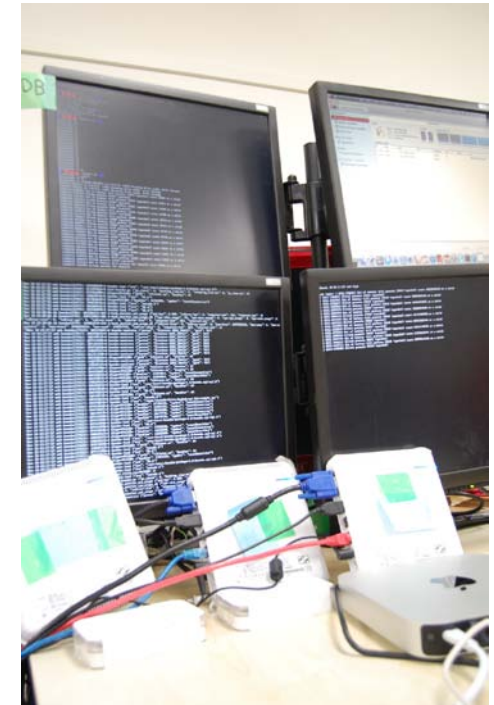
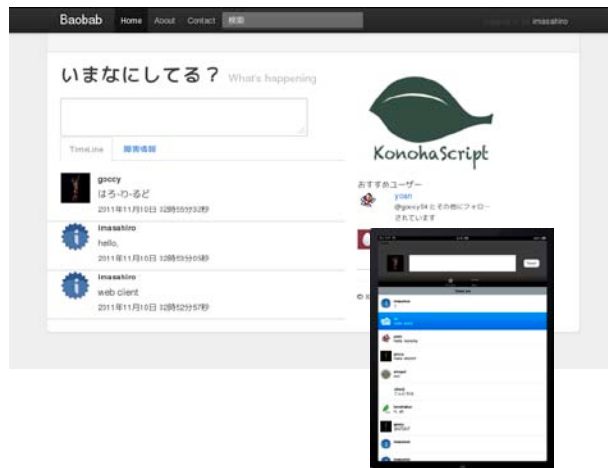


Konohaによる D-Script 参照実装



D-Script 実験環境

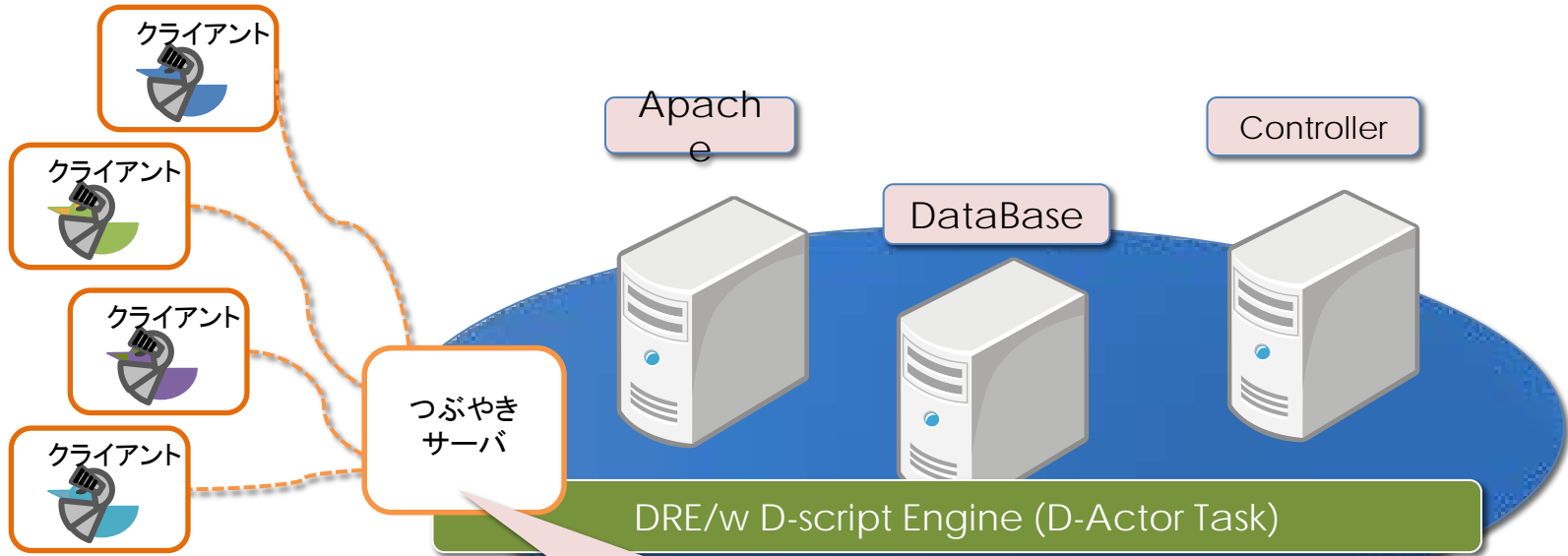
モバイル端末を含めた
Mini-Twitter アプリケーション



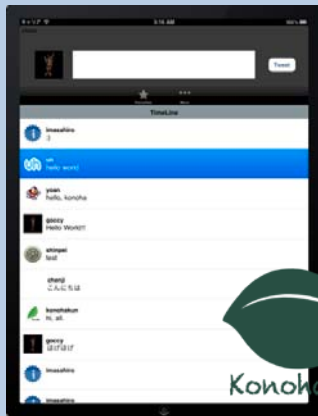
↑
フォルト・インジェクター
異常負荷 (DS-Bench, 東大)
システムエラー (FaultVM, 筑波大)

↑
D-Script
モニター・診断スクリプト
障害回復スクリプト

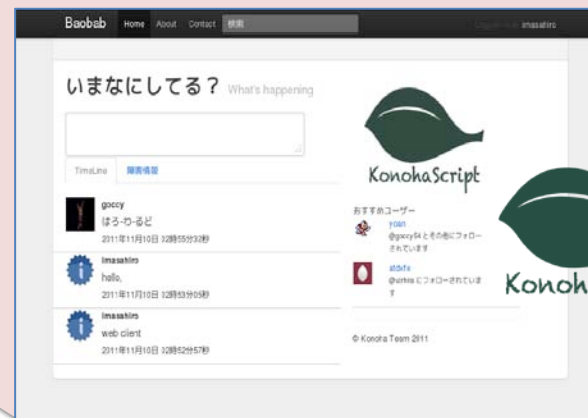
ET 2011 デモ展示



クライアント
インターフェイス

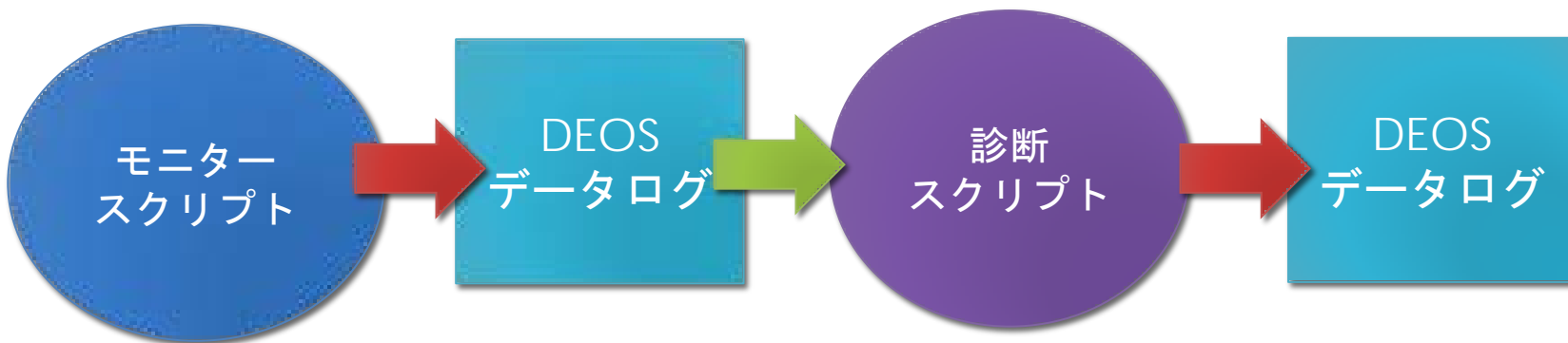
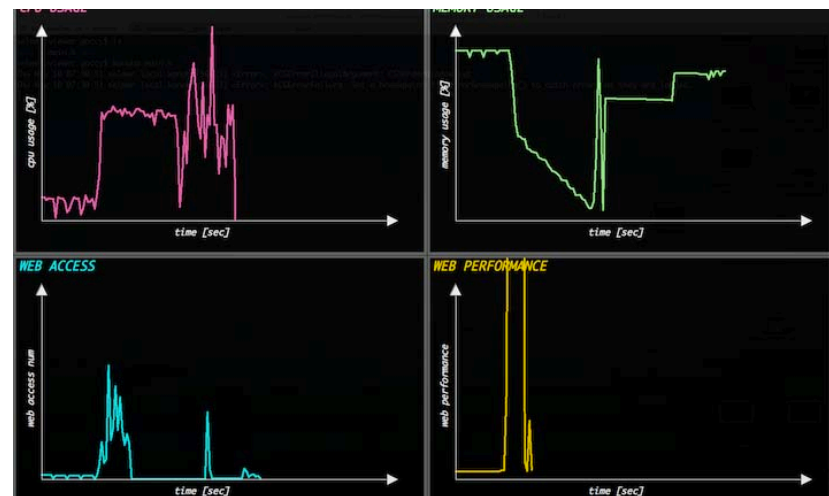


サーバインターフェイス



D-Scriptの例: モニター開発

- D-Script ログライブラリ
 - DEOS データログ形式
 - クロスドメイン/トレース
 - スナップショット
 - ランダムアクセス



今後のロードマップ

- 基本機能 (参照実装)
 - オープンソース公開
 - KonohaScript 1.0
 - <http://www.konohascript.org/>
- DEOS 拡張機能
 - 「DEOS 普及促進委員会」から提供
 - D-Script DSL, セキュリティ機能, モニター連携
 - D-Script 仕様書



KONOHASCRIPIT DOCUMENT / DOWNLOAD

About KonohaScript

Scripting languages, such as JavaScript and Ruby, are booming and widely accepted in industries. Their dynamic features are very nice and offer you flexibility when you start coding. But, as your scripts grow larger, you would suffer from the lack of static programming features.

Do you want to rewrite your scripts to C++ or Java? Or, are you still trying to maintain bloated scripts by hands? Neither is so attractive answer.

KonohaScript is newly designed to deliver both flexible scripting and static programming with one single language. Your scripts are programmed smoothly, and serve well with a long-term maintenance.

Features

KonohaScript is a statically-typed object-oriented scripting language. Dynamic features are carefully added without any loss of good static features.

- lightweight objects
- dynamic code loading (a.k.a eval)
- gradual typing (controlled dynamic type)
- partial execution of incomplete programs
- runtime alternation of existing classes
- extensible URN literals