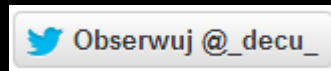


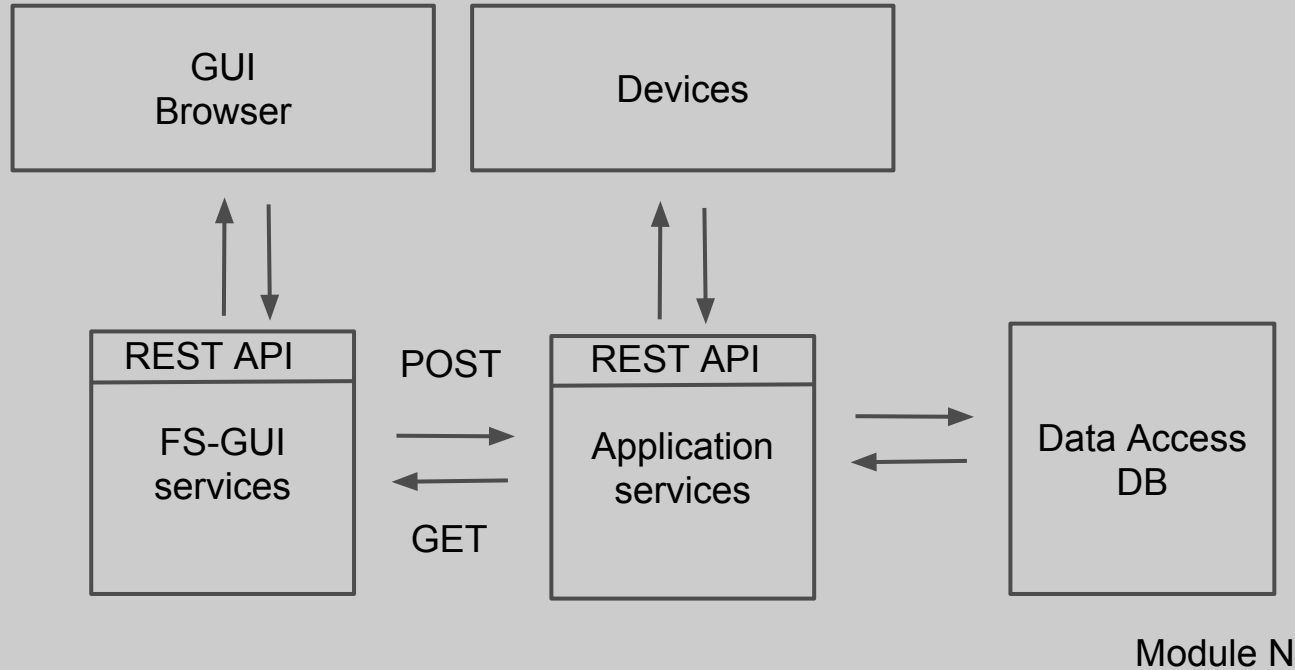
# ATF

Nietypowe podejście do automatyzacji  
w systemie rozproszonym

Daniel Dec  
[ddec@future-processing.com](mailto:ddec@future-processing.com)



# Architektura SUT



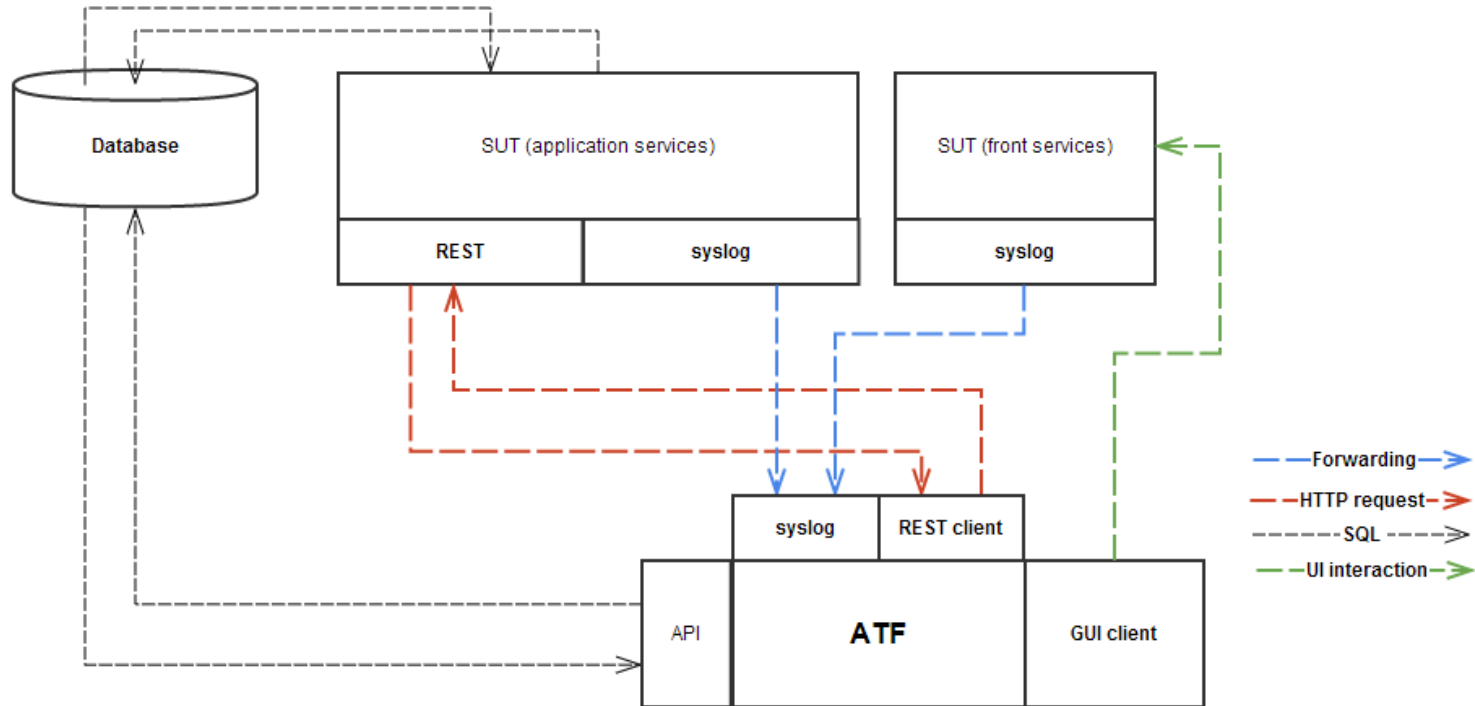
# Założenia ATF

- zewnętrzne narzędzie (instalacja)
- dane wejściowe/wyjściowe
- zbieranie danych stanu systemu
- bezstanowość
- prędkość wykonywania
- synchroniczność

# Założenia ATF

- DSL (Domain Specific Language) - warstwa abstrakcji
- reużywalność kroków
- unifikacja i generowanie transakcji
- wynik testu przez porównanie stanów

# Wysokopoziomowa architektura





# Raport testów

- elementy raportu
- statusy wyniku
- różnicowy
- aktualizacja danych referencyjnych

(przykłady)

# Retrospektywa #1

- szersza perspektywa
- przezroczystość systemu
- utrzymywanie danych referencyjnych
- prędkość wykonywania/deterministyczność
- typy asercji (ewolucja)



# Reużywalność - przykład Gherkin

```
1 Feature: Stock entity updating
2 #-----
3 # SM_037_TPRO - Transactions relating to Travel Cards
4 #-----
5
6 @external_id=HEL-3288 @dg2 @br=██████████ @author=██████████ @modified_by=██████████
7 Scenario Outline: SM - Decreasing stock entity by TravelCardIssuing transaction
8   Given Stock Entity with unique ID has been created with thresholds set to "100"
9     And quantity in Stock Entity with ID is updated to following "<travelCardsCount>"
10    When "<amount_of_tc>" transactions of type "TravelCardIssuing" with parameter "stockEntityIdentifier" set to Stock
11      Entity ID have been inserted
12      Then Travel Card quantity in the Stock Entity with ID is equal "<travel_card_quantity>"
13 Examples:
14   | travelCardsCount | travel_card_quantity | amount_of_tc |
15   | 1                 | 0                     | 1             |
16   | 1                 | 0                     | 10            |
17   | 2                 | 1                     | 1             |
```

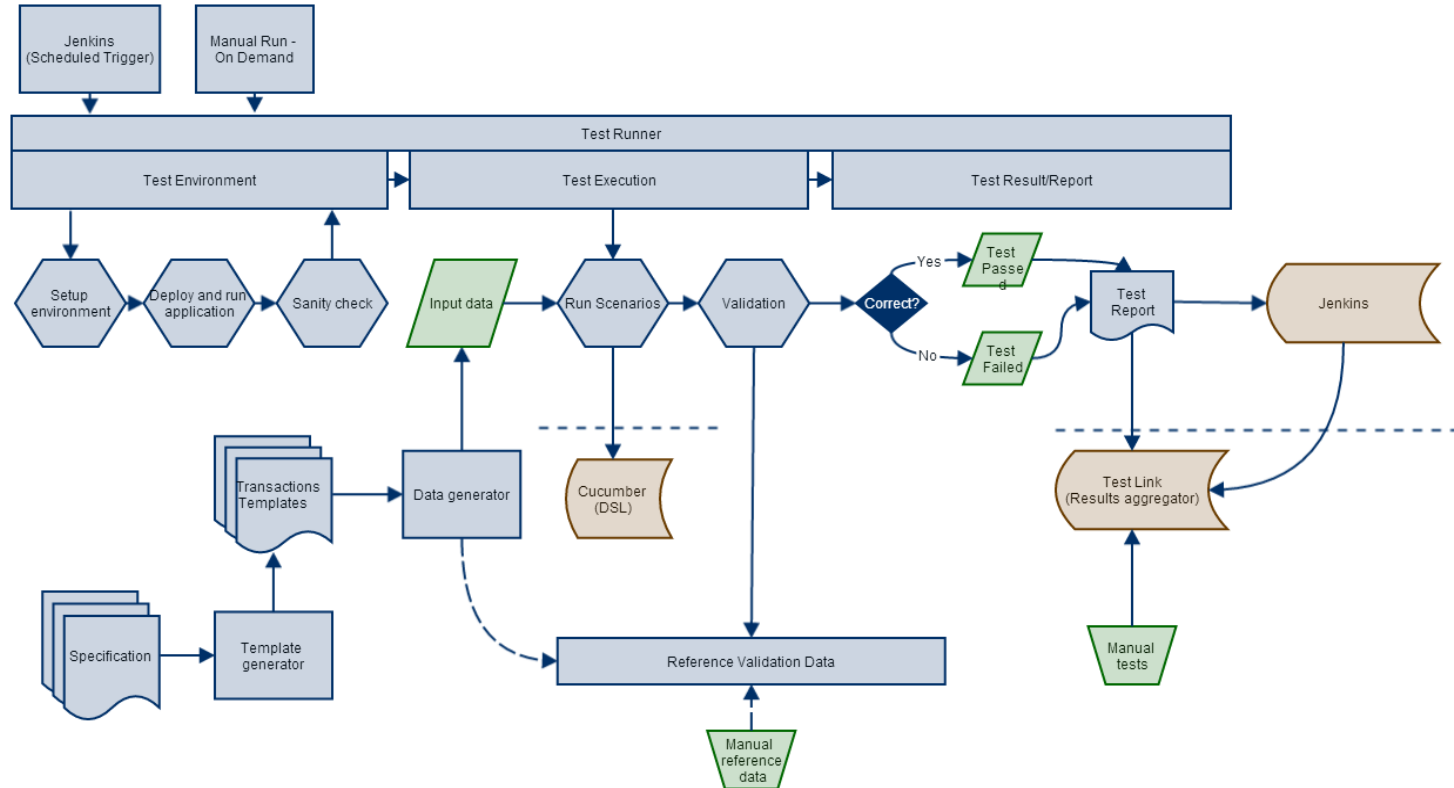
# Generatory

- szablony .ejs
- modele danych
- unifikacja
- generatory zapytań REST

# Typy i rodzaje testów

- testy modułowe
- testy integracyjne
- testy przekrojowe (e2e)
- sanity checks
- testy GUI

# Wysokopoziomowy widok



# Ciągłe testowanie

- CI
- zarządzanie środowiskami testowymi
- typy buildów
- provisioning środowisk

# Retrospektywa #2

- elastyczność i możliwość rozszerzania
- agregacja wielu narzędzi
- narzędzie dla całego zespołu
- współdzielony kod
- sterowanie wykonaniem testów (tagi)
- całościowe rozwiązanie - wsparcie dla wielu poziomów testów

# Główne narzędzia

Python, VM tools, Jenkins, TestLink, SQL database tracking, Sublime (+plugin intellisense), lettuce, Selenium, RedHat (...)

# Pytania?



 Future Processing