

# Ανάλυση & σχεδίαση υπηρεσίας νέφους με χρήση μικροϋπηρεσιών για διαπλατφορμικές φορητές υπολογιστικές συσκευές

Παρουσίαση διπλωματικής εργασίας

Βασιλούδης Βασίλειος Ιούλιος, mai16004  
(Μεταπτυχιακός Φοιτητής)

Επιβλέπων Καθηγητής: Χατζηγεωργίου Αλέξανδρος



Πρόγραμμα Μεταπτυχιακών Σπουδών

**Τμήμα Εφαρμοσμένης Πληροφορικής**  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

Φεβρουάριος 2018

# Περιεχόμενα

- ✓ Σκοπός μελέτης
- ✓ Βασικές αρχές & χαρακτηριστικά μικροϋπηρεσιών
- ✓ Τύποι διαπλατφορμικών τεχνολογιών κινητών συσκευών
- ✓ Παρουσίαση υλοποιημένου συστήματος
- ✓ Συμπεράσματα
- ✓ Μελλοντικές επεκτάσεις



# Σκοπός μελέτης



# Σκοπός της μελέτης

Διερεύνηση αρχιτεκτονικών προτύπων μικροϋπηρεσιών συνδυαστικά με cross-platform τεχνολογίες για κινητές συσκευές με στόχο:

- Παρουσίαση πλεονεκτημάτων & μειονεκτημάτων
- Μελέτη ευελιξίας και συντηρησιμότητας έναντι των συμβατικών μοντέλων.
- Εφαρμογή γνωστών καλών πρακτικών & προτύπων σχεδίασης

Δημιουργία “proof of concept” ολοκληρωμένου συστήματος

- Υλοποίηση υπηρεσίας νέφους με χρήση μικροϋπηρεσιών
- Υλοποίηση εφαρμογής πελάτη κινητής συσκευής με υβριδικές τεχνολογίες (cross-platform)
- Εγκατάσταση υποστηρικτικών υπηρεσιών monitoring
- Διαρκής ενσωμάτωση/εγκατάσταση (CI/CD)
- Διανομή μέσω docker containers

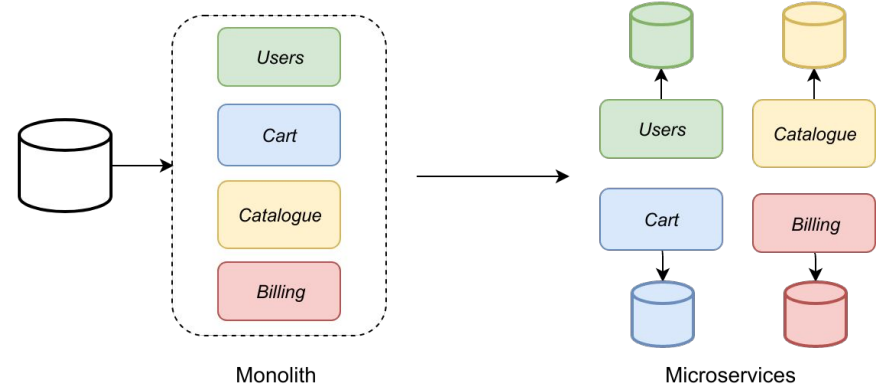


# Βασικές αρχές & χαρακτηριστικά μικροϋπηρεσιών



# Τι είναι οι μικροϋπηρεσίες;

- Διαχωρισμός εφαρμογής σε μικρές αυτόνομες υπηρεσίες
- Ενθαρρύνει την παράλληλη ανάπτυξη & ανεξάρτητη εγκατάσταση
- Αποκέντρωση δεδομένων
- Δημιουργία κουλτούρας αυτοματισμού
- Απομόνωση αποτυχιών



# Σύγκριση με μονολιθικό μοντέλο

## Μικροϋπηρεσίες

- + Ισχυρά όρια αρμοδιοτήτων
- + Ανεξάρτητη εγκατάσταση
- + Πολυμορφία τεχνολογιών
- + Κλιμάκωση
- Πολυπλοκότητα
- Απόδοση (καθυστερήσεις δικτύου κτλ)

## Μονόλιθος

- + Απλούστερη εγκατάσταση
- + Διαχείριση transactions
- + Ευκολότερο monitoring & testing
- Μεγάλο codebase
- Περιορισμένη κλιμάκωση
- Χρονοβόροι κύκλοι ανάπτυξης
- Υψηλή σύζευξη χωρίς δυνατότητα απομόνωσης



# Τύποι διαπλατφορμικών τεχνολογιών κινητών συσκευών





# Τύποι διαπλατφορμικών τεχνολογιών κινητών συσκευών

- Ιστού ή (Progressive Web Applications)
- Υβριδικές
- Javascript/Native ή (Native App Shell)
- Cross-compiled



# Ιστού

- + Χρόνος ανάπτυξης
- + Ευελιξία αναβάθμισης (αυτόματα μέσω του .manifest)
- Περιορισμένη πρόσβαση συσκευής (μέσω browserAPIs)
- Χρήση μόνο μέσω browser



# Υβριδικές

- + Ενιαίος κώδικας
- + Συνδιασμός web τεχνολογίας με χρήση plugins για πρόσβαση σε κάθε πλατφόρμα
- Μειωμένες επιδόσεις (λόγω χρήσης WebView)
- Αυξημένο μέγεθος εκτελέσιμου



# Javascript/Native

- + Ενιαίος κώδικας
- + Καλές επιδόσεις (λόγω χρήσης Native UI)
- Μερική επαναχρ/ποίηση (διαφοροποιήσεις κώδικα UI ανά πλατφόρμα)
- Μειωμένες επιδώσεις σε περιπτώσεις υψηλού υπολογιστικού φόρτου



# Cross-compiled

- + Πλήρης υποστήριξη υποσυστημάτων συσκευής
- + Επιδόσεις σχεδόν ως Native
- + Πλήρης επαναχρ/ποίηση\* κώδικα
- Η επαναχρ/ποίηση εξαρτάται άμεσα από τις παρεχόμενες βιβλιοθήκες
- Περιορισμένες open-source βιβλιοθήκες
- Μεγαλύτερο μέγεθος εκτελέσιμου
- Όχι για εφαρμογές με έντονα γραφικά



**RUBYMOTION**



# Σύγκριση με native πρακτικές για κινητές συσκευές

## Cross-platform

- + Επαναχρ/ποίηση κώδικα
- + Μικρότερος χρόνος & κόστος ανάπτυξης
- + Μεγαλύτερη γκάμα γλωσσών προγρ/μου
- Χαμηλότερες επιδόσεις
- Περιορισμένη χρήση υποσυστημάτων συσκευής

## Native

- + Πλήρη πλεονεκτήματα συσκευής
- Αποκλειστικό SDK ανά πλατφόρμα
- Πολλαπλή ανάπτυξη ανά λειτουργικό
- Αυξημένο κόστος ανάπτυξης/συντήρησης



# Παρουσίαση υλοποιημένου συστήματος



Vento

Username

Password

[Forgot your password?](#)

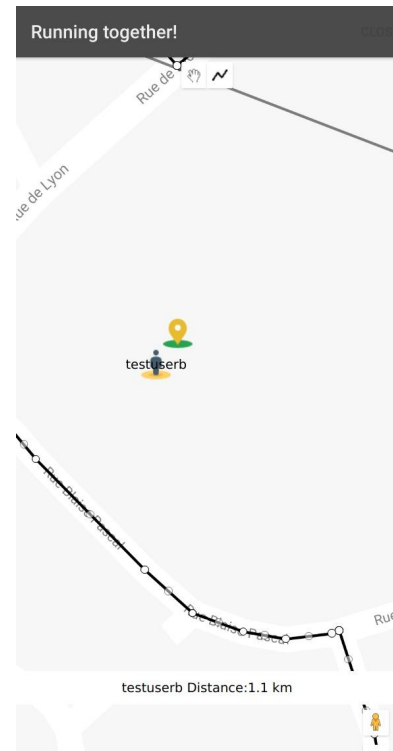
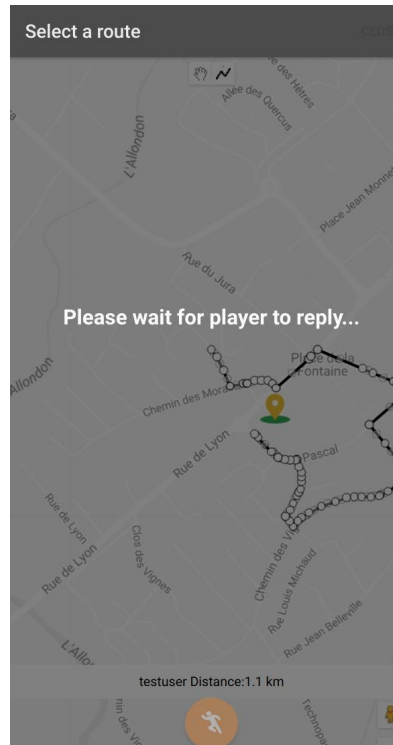
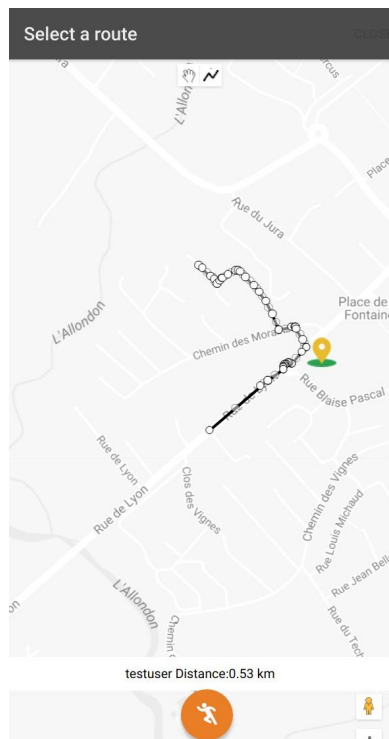
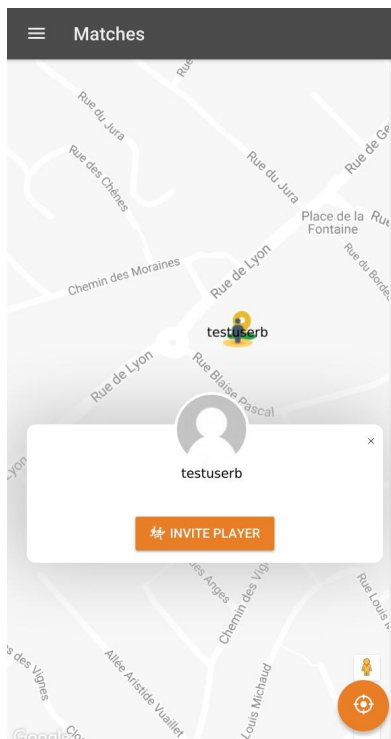
LOGIN

Don't have an account yet? [Sign up](#)





# Εφαρμογή



# Υπηρεσία νέφους

## Προδιαγραφές συστήματος

- ✓ Αποκεντρωμένη διαχείριση δεδομένων
- ✓ Αυτοθεραπευόμενο σύστημα
- ✓ Εύκολη επανεγκατάσταση
- ✓ Ενοποιημένο API
- ✓ Κλιμακούμενο



mongoDB



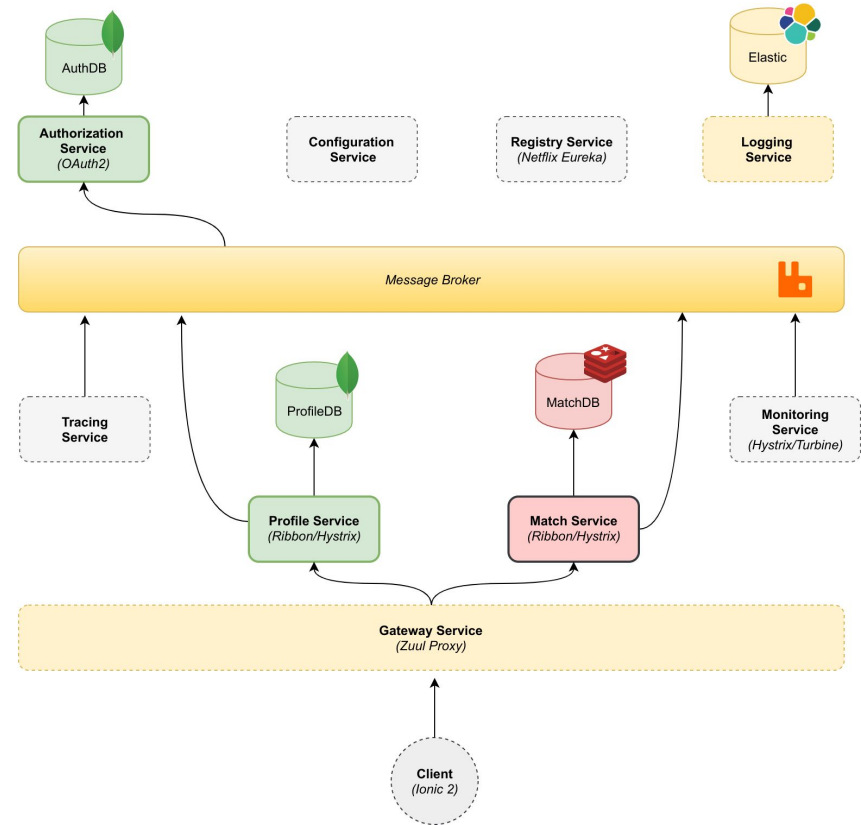
# Πρότυπα σχεδίασης

- Εξωτερικευμένη πηγή ρυθμίσεων (Externalized configuration)
- Κανάλι μηνυμάτων (Message bus / Messaging)
- Πύλη Api (Api gateway)
- Μητρώο υπηρεσιών (Service registry)
- Διακόπτης (Circuit breaker)
- Κατανεμημένος εντοπισμός (Distributed tracing)



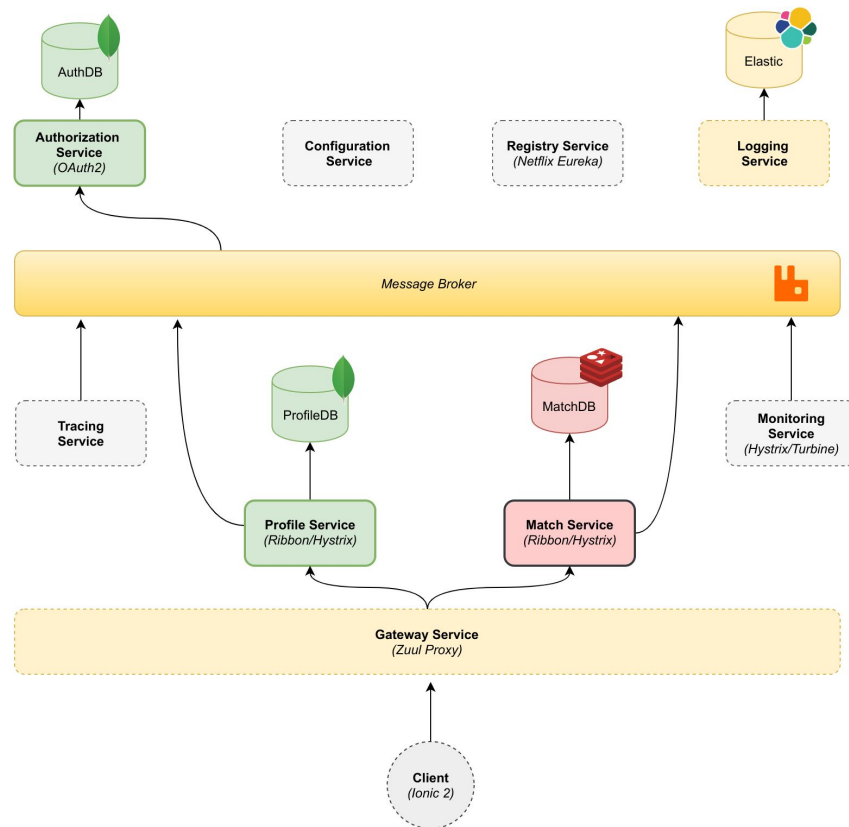
# Υπηρεσίες κορμού

- **Profile Service**
  - Αποθήκευση προφίλ χρηστών
  - MongoDB
- **Auth service**
  - Αυθεντικοποίηση με OAuth2
  - MongoDB
- **Match service**
  - Εύρεση Geolocation χρηστών
  - Σύνδεση μέσω Websocket
  - Redis



# Υπηρεσίες υποδομής

- **Configuration service**
  - Ενιαίο σημείο ρυθμίσεων υπηρεσιών
- **Gateway service**
  - Proxy/Load balancer
  - Χρήση Netflix Zuul/Ribbon
- **Monitoring service**
  - Circuit Breaker
  - Χρήση Netflix Hystrix



# Υπηρεσίες υποδομής

## ➤ Registry service

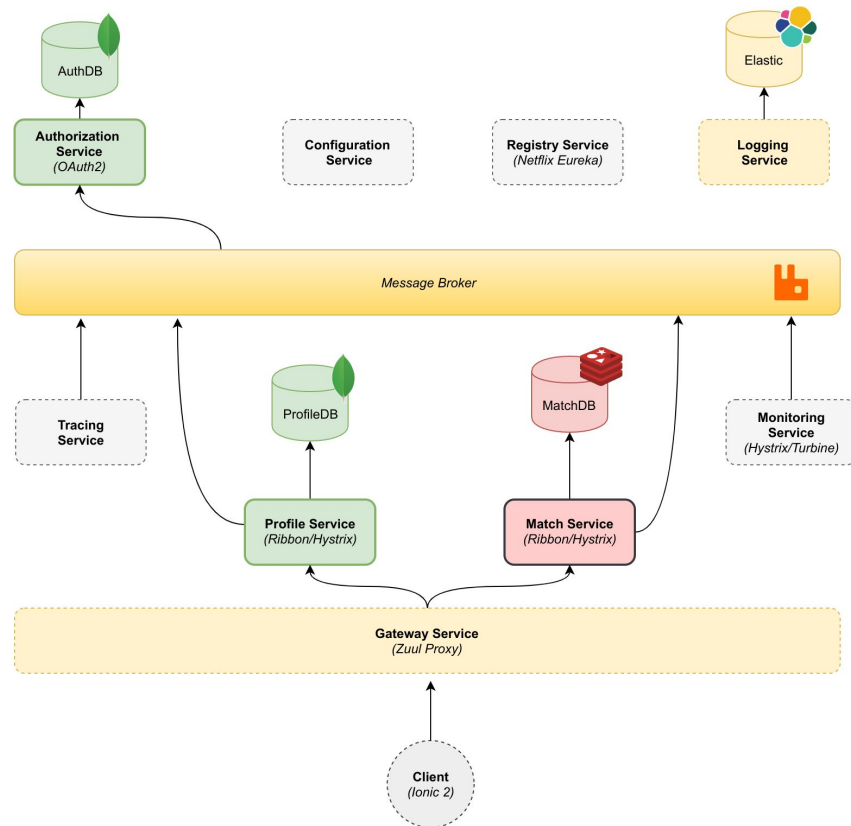
- Μητρώο ενεργών υπηρεσιών
- Netflix Eureka

## ➤ Logging service

- Κεντρική συλλογή logfiles
- Elastic/Kibana/Logstash

## ➤ Tracing service

- Δειγματοληψία κλήσεων για bottlenecks (Correlation ids)
- Zipkin



# Εφαρμογή πελάτη

- Hybrid τεχνολογία (Κοινός κώδικας ανάμεσα στις πλατφόρμες)
- Ionic framework (v3)
  - Html5
  - Angular 4
  - Typescript
  - Scss
  - Cordova
- Χρήση plugins για πρόσβαση σε υποσυστήματα (Geolocation, StepCounter κ.α)

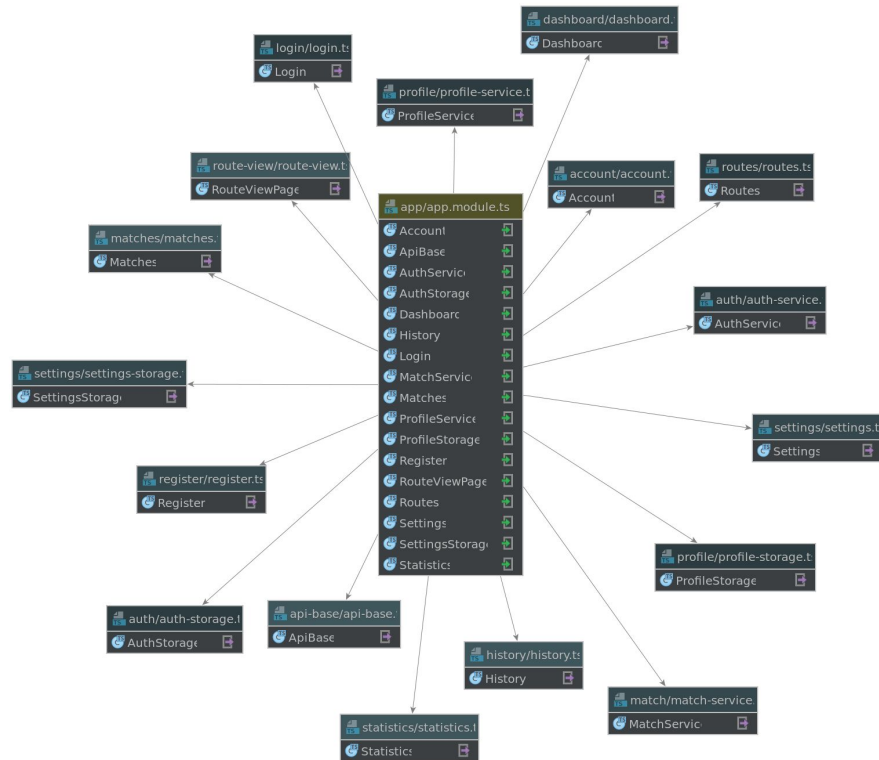


# Εφαρμογή πελάτη

## ➤ Providers

- Services (Business logic)
- Storages (Αποθήκευση σε LocalStorage)

## ➤ Components (γραφικό περιβάλλον χρήστη)



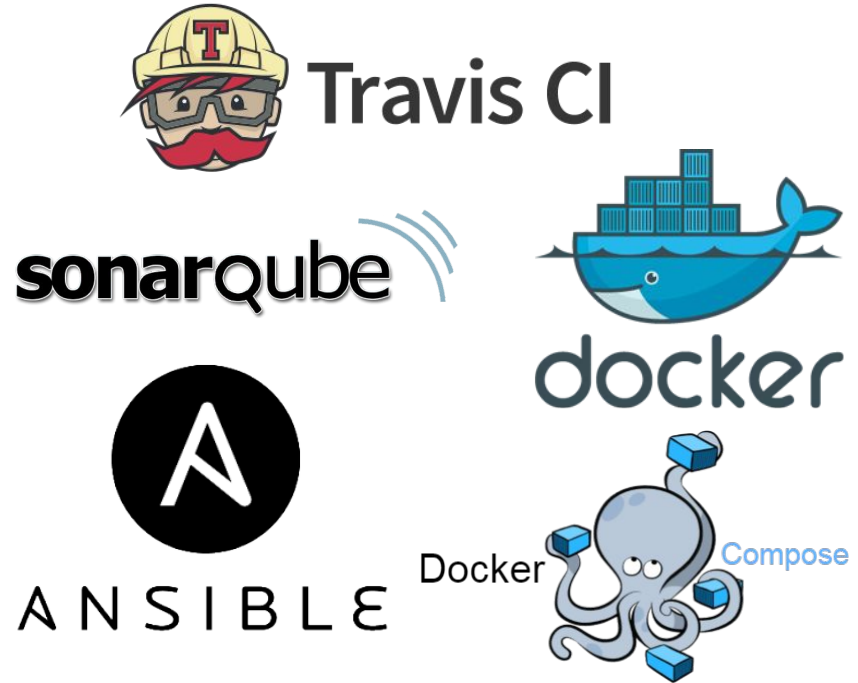
Powered by yfiles





# Συνεχής ενσωμάτωση/εγκατάσταση

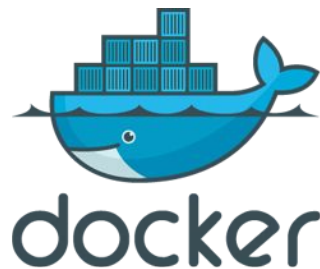
- TravisCI
- SonarCloud (SonarQube)
- Docker
- Docker Compose
- Ansible



# Συνεχής ενσωμάτωση/εγκατάσταση

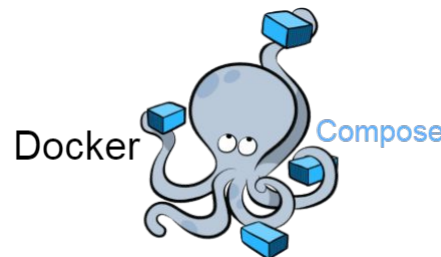
## Docker

- Απομονωμένο περιβάλλον
- Ενσωμάτωση runtime dependencies σε container
- Rapid deployment



## Docker Compose

- Ενορχήστρωση πολλαπλών docker containers



# Συνεχής ενσωμάτωση/εγκατάσταση

## Ansible

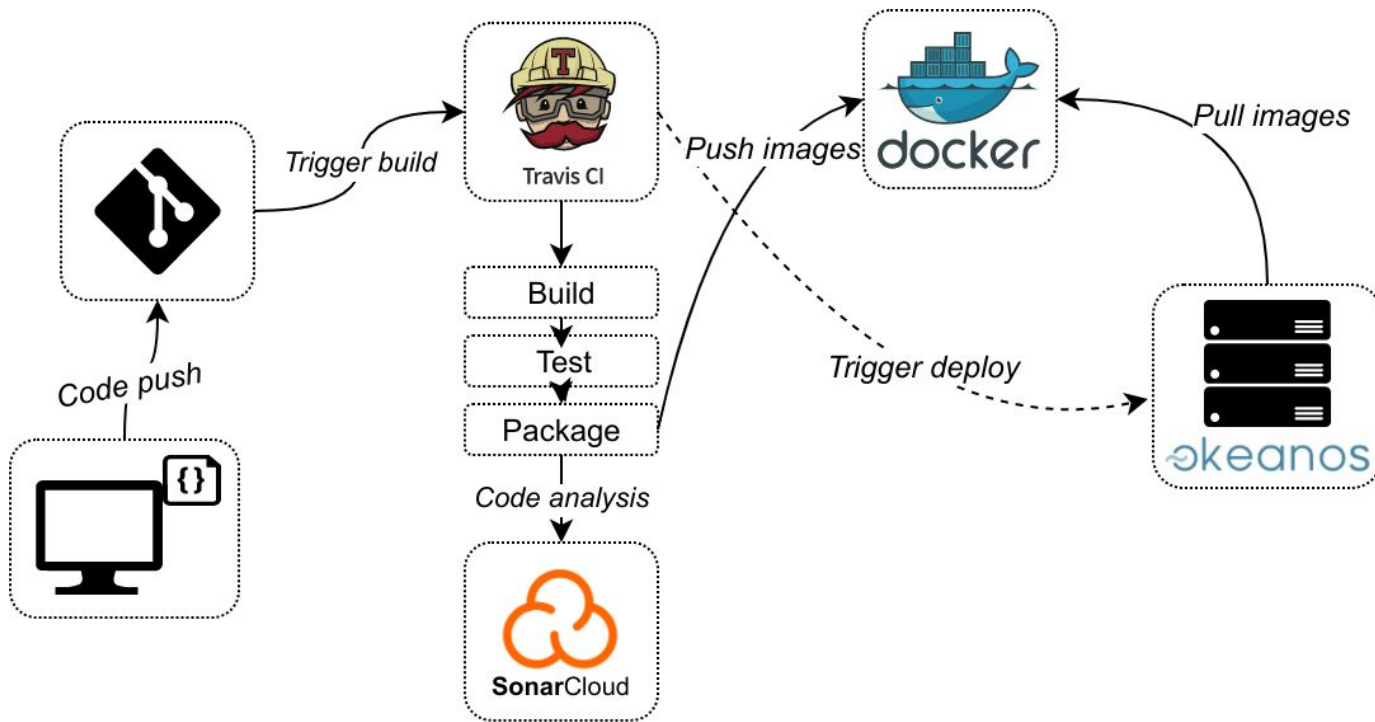
- Push-based εργαλείο deployment
- Agentless
- Ενσωματωμένη υποστήριξη docker μέσω modules (docker\_container,docker\_image)



A N S I B L E



# Διάγραμμα εγκατάστασης



# Συνολικά συμπεράσματα & μελλοντικές επεκτάσεις



# Συμπεράσματα

- Προγραμματιστική ευελιξία ανά ομάδα και καλύτερος διαχωρισμός της επιχειρησιακής λογικής ανά υπηρεσία
- Αναγκαία η χρήση αυτοματισμών εγκατάστασης (DevOps)
- Εύκολη κλιμάκωση με δημιουργία πολλαπλών instances
- Ανεξάρτητο τεχνολογιών μελλοντικά λόγω εύκολης αντικατάστασης μεμονωμένων υπηρεσιών
- Μεγάλο κόστος ενορχήστρωσης με πολλές υπηρεσίες για ομαλή λειτουργία
- Προοδευτική μετατροπή μονόλιθου σε μικροϋπηρεσίες



# Συμπεράσματα

- Ικανοποιητική επαναχρησιμοποίηση κώδικα
- Οι υβριδικές τεχνολογίες υποφέρουν από χαμηλές επιδόσεις σε σχέση με NativeView Renderers ακόμη και σε δυνατότερες συσκευές
- Ανάλογα με την φύση του έργου προτείνεται ανάλογη τεχνολογία
  - πχ. Native για εφαρμογές υψηλού φόρτου
  - ή Cross-compiled/Javascript-Native για απλές εφαρμογές



# Μελλοντικές επεκτάσεις

- Εμβάθυνση προτύπων σχεδίασης μικροϋπηρεσιών
- Εφαρμογή σε ένα πραγματικό σενάριο μεγαλύτερης κλίμακας με σκοπό επίλυσης των προβλημάτων





Ευχαριστώ για την προσοχή σας!  
Ερωτήσεις;

