

Specifica Tecnica v1\_0\_0 2017-04-21 esterno Questo documento descrive l'architettura generale del prodotto Monolith che verrà sviluppato dal gruppo Obelix.

## Diario delle revisioni

Modifica	Autore e Ruolo	Data	Versione
Approvazione documento	Nicolò Rigato Responsabile	2017-03-09	1.0.0
Verifica del documento	Silvio Meneguzzo Verificatore	2017-03-09	0.1.0
Stesura sezione Resoconto delle attività di verifica	Tomas Mali Verificatore	2017-03-05	0.0.6
Stesura sezione Gestione amministrativa della revisione	Riccardo Saggese Verificatore	2017-03-04	0.0.5
Stesura sezione Strategie di Verifica	Tomas Mali Verificatore	2017-02-28	0.0.4
Stesura sezione Definizione obiettivi di qualità	Riccardo Saggese Verificatore	2017-02-27	0.0.3
Stesura sezione Introduzione	Tomas Mali Verificatore	2017-02-26	0.0.2
Creazione template	Nicolò Rigato Responsabile	2017-02-25	0.0.1

## Indice

## Elenco delle tabelle

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento ha come scopo quello di definire la progettazione ad alto livello per il prodotto Monolith. Verrà presentata l'architettura generale secondo la quale saranno organizzate le varie componenti software e i Design Pattern utilizzati nella creazione dell'SDK, delle bolle predefinite e della demo. Verrà inoltre dettagliato il tracciamento tra le componenti software individuate ed i requisiti.

## 1.2 Scopo del prodotto

Lo scopo del prodotto è quello di permettere la creazione di bolle interattive, che dovranno funzionare nell'ambiente Rocket.chat. Queste bolle permetteranno di aumentare l'interattività tra gli utenti della chat e aggiungeranno nuove funzionalità accessibili direttamente dalla conversazione senza il bisogno di ricorrere all'apertura di applicazioni diverse. Il sistema offrirà agli sviluppatori un set di API per creare e rilasciare nuove bolle e agli utenti finali la possibilità di usufruire di un insieme di bolle predefinite.

## 1.3 Glossario

Al fine di evitare ogni ambiguità di linguaggio e massimizzare la comprensione dei documenti, i termini che necessitano di essere chiariti saranno scritti in corsivo e marcati con una |G| in pedice alla prima occorrenza e saranno riportati nel Glossario.

## 1.4 Riferimenti

### 1.4.1 Normativi

- **Norme di Progetto:**  
NormediProgetto\_v1.1.0
- **Capitolato d'appalto C5:**  
<http://www.math.unipd.it/~tullio/IS-1/2016/Progetto/C5.pdf>
- **Analisi dei Requisiti:**  
Analisi dei Requisiti\_v1.0.0

### 1.4.2 Informativi

- **Slide del corso di Ingegneria del Software:**  
<http://www.math.unipd.it/~tullio/IS-1/2016/>

## 1.5 Tecnologie Utilizzate

In questa sezione verranno descritte le tecnologie su cui si basa lo sviluppo del progetto. Per ognuna di esse, verranno indicati l'ambito di utilizzo della tecnologia, i vantaggi e gli svantaggi che ne derivano. Alcune delle tecnologie che saranno usate sono richieste come requisito dal capitolato scelto.



### 1.5.1 Javascript 6th edition (ECMA SCRIPT 6)

JavaScript è un linguaggio di scripting orientato agli oggetti e agli eventi. È comunemente utilizzato nella programmazione Web lato client per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite l'uso di funzioni di script invocate da eventi innescati in vari modi dall'utente sulla pagina web in uso.

Come richiesto dal capitolato, per la realizzazione di Monolith, deve essere utilizzato Javascript 6th edition (ECMA SCRIPT 6).

#### **Licenza:**

Non esiste una sola implementazione perché ECMAScript (o ES) è un linguaggio di programmazione standardizzato e mantenuto da Ecma International nell'ECMA-262 ed ISO/IEC 16262.

#### **Vantaggi:**

- Gestione degli eventi asincroni tramite le promises
- Possibilità di dichiarare classi
- Supporto per le costanti(*const*)
- Possibilità di isolare la definizione di variabili ad un blocco (*let*)
- Possibilità di isolare lo scope di una funzione usando blocchi delimitati da parentesi graffe() come ambienti isolati (vs closure)
- Uso di sintassi più espressiva per scrivere le funzioni anonime (*Arrow Functions*)

#### **Svantaggi:**

- Il supporto di ES6 da parte dei browser è ancora incompleto
- L'assenza di tipizzazione potrebbe ostacolare la valutazione della correttezza del codice

### 1.5.2 Meteor

Meteor è un framework web JavaScript libero e open source per lo sviluppo di applicazioni web e mobile. È una piattaforma basata su Node.js. Meteor utilizza, dunque, JavaScript sia lato client che lato server.

#### **Licenza:** MIT

La licenza MIT è una delle licenze più permissive nel panorama open source. In modo più esplicito dichiara i diritti dati all'utente finale, incluso il diritto di utilizzare, copiare, modificare, incorporare, pubblicare, distribuire, sotto-licenziare, e/o vendere il software.

#### **Vantaggi:**

- Integrazione con diverse tecnologie utilizzate nello sviluppo web:
  - React
  - MongoDB

- Isomorfismo: il codice javascript scritto funziona in modo trasparente sul client (browser), sul server (Node.js) o in entrambi i mondi
- Ecosistema e modularità: la comunità di Meteor è molto attiva e molte funzionalità client o server potrebbero già essere pacchettizzate dal package manager ufficiale.

**Svantaggi:**

- Inizialmente sconosciuto ai membri del gruppo.

### 1.5.3 Mongo DB

MongoDB è un database NoSQL orientato ai documenti, basato sul formato JSON per la memorizzazione e la rappresentazione dei dati. È distribuito come software libero open source.

**Licenza:** GNU AGPL v3.0

È una licenza di software libero pubblicata da Free Software Foundation. È simile alla capostipite GNU GPL.

**Vantaggi:**

- È più flessibile di un database SQL e facilita la rappresentazione su un modello ad oggetti
- Supporta ricerche per campi, intervalli e regular expression. Le query possono restituire campi specifici del documento e anche includere funzioni definite dall'utente in JavaScript.
- Qualunque campo in MongoDB può essere indicizzato

**Svantaggi:**

- Inizialmente sconosciuto ai membri del gruppo.

### 1.5.4 HTML5

HTML5 è un linguaggio di markup per la strutturazione delle pagine web.

**Vantaggi:**

- Codice più pulito e sintassi semplificata rispetto alle versioni precedenti
- Interattività senza l'ausilio di plugin esterni valida per diversi formati multimediali
- Semantica intuitiva grazie ai nuovi TAG di formattazione
- Introduzione della geolocalizzazione, dovuta ad una forte espansione di sistemi operativi mobili
- Sistema più efficiente alternativo ai normali cookie chiamato Web Storage

**Svantaggi:**

- Non tutti i browser supportano HTML5.

### 1.5.5 SCSS

SCSS è una sintassi per i fogli di stile introdotta da Sass 3 (Syntactically Awesome StyleSheets). È un'estensione del CSS .

**Vantaggi:**

- Possibilità di utilizzare variabili
- Possibilità di creare funzioni
- Possibilità di organizzare il foglio di stile in più file
- Compatibilità completa con la sintassi del CSS

**Svantaggi:**

- Sintassi più complessa.

### 1.5.6 React

React è una libreria Javascript open source che permette di costruire interfacce utente.

**Vantaggi:**

- Semplificazione della realizzazione di interfacce UI dinamiche che possono reagire ai cambiamenti di dati in maniera autonoma attraverso opportuni componenti
- Possibilità di usare le viste per creare codice più facile da comprendere e su cui è più semplice effettuare il debug

**Svantaggi:**

- 

### 1.5.7 Node.js

**Vantaggi:**

- 

**Svantaggi:**

- 

### 1.5.8 Rocket.chat