
紹介「**Master of
Cloud/Mobile アプリエン
지니어への道**」

<http://goo.gl/iVJtTJ>

QOP GATE

@shin1ogawa

Google Developer Expert / Cloud Team

- Google App Engine
- Google Apps API



Google Developer Experts - <https://developers.google.com/experts/members/>

株式会社トツプゲート_CTO

- Google App Engine, Google Apps API
 - Java / JavaScript / Golang / Objective-C
 - Enterprise Web Applications, iOS, Android



<https://plus.google.com/+ShinichiOgawa>

結論:何を学ばせたいか？

結論:何を学ばせたいのか？

プログラムを書けるだけのプログラマではなく、アプリケーションを構築できるプログラマを技術、姿勢の面で育成する

- Issue Tracker、ソースコード管理
 - Maven, Grunt等のビルドツール
 - Unit testツール
 - 上記を使ったCIの運用、自動化の重要性
 - PaaS上で稼働するアプリケーションの開発
 - ブラウザ上で稼働するアプリケーションの開発
 - 継続的な学習の重要性
-

最近のアプリケーション開発事情

最近のアプリケーション開発事情

Platform

- Enterprise向け
 - Server: Cloud, On-premises
 - Client: Web browser(PC), Web browser (Smart device), Smart device
- Consumer向け
 - Server: Cloud, On-Premises
 - Client: Smart device, Web browser(Smart device), Web browser(PC)

for Enterpriseが、for Consumer の流れを追う方向

最近のアプリケーション開発事情

Platform

Server側はどんどんCloudに移行するのは間違いないので...

- Cloud(API) + Web Browser(JavaScript) on PC
- Cloud(API) + Web Browser(JavaScript) on Smart device
- Cloud(API) + Native(Java, iOS) on Smart device

これらのパターンにはまるものが多くなる

最近のアプリケーション開発事情

Platform

Server側はどんどんCloudに移行するのは間違いないので...

- Cloud(API) + Web Browser(JavaScript) on PC
- Cloud(API) + Web Browser(JavaScript) on Smart device
- Cloud(API) + Native(Java, iOS) on Smart device
- *BaaS(開発無) + Native(Java, iOS) on Smart device*

これらのパターンにはまるものが多くなる

Cloud, Mobileの影響

Cloud, Mobileの影響

困る(?)

- Cloudの「安い」イメージにより、低コスト短工期な流れはこれまで以上に加速
 - 低コストに対応するための、単価の安い**Poor**なエンジニアが増えてしまっている
 - ユーザが**Consumer**側の立場として、リッチな体験を普通にしている
 - **Framework**やツールを使った、画一的な**Framework**だけでは良い**UX**を提供できない
 - アプリ固有の**UX**をアプリごとに設計実装しなければ満足は得られない
-

Cloud, Mobileの影響

良くなっている

- インフラの初期コスト、運用コストが下がったため、小規模な組織や個人でも、全世界に向けたアプリケーションを開発運用できるようになっている
 - マーケットプレイスが普通になり、ユーザに直接リーチする機会が劇的に増えている
 - ユーザが色々な状況(場所時間)で頻繁にデバイスを利用できるので、アプリを使ってもらえる機会も増えている
-

Cloud, Mobileの影響

困難を克服しつつ、チャンスは利用したい

- 少人数、1人、分散した場所での作業でも、さくさくとアプリケーション構築のサイクルを回せる人員を育てる
 - 1人でアプリケーションを構築できるのが良い
 - 実際そのような人材が切に求められている
 - 少人数チームで生産性を上げるには？
 - CIを前提に、ビルドやUnit Test等のプロセスを、最大限の範囲で自動化する。
 - ユーザが触るフロントエンドがキモなので、バックエンド側の工数を削減する。
-

開発者を助けるツール類

Tools

自動化

- 人間が作業すると必ずミスをするので、自動化できるところは自動化する
 - 環境
 - ソースコード管理、チケット駆動
...を自動的に連動させる
 - アプリケーション
 - コンパイル、テスト、パッケージング、デプロイ
...を自動化する
-

Tools

可視化

- 関係者の行動、関係するリソースへの変更、を可視化する
 - 環境
 - ソースコード管理、チケット駆動
...を自動的に配信する
 - アプリケーション
 - コンパイル、テスト、パッケージング、デプロイ
...の結果も自動的に配信する
-

Tools

！注意

- 自動化も可視化も、できていないプロジェクトは全くできていない。
そういう環境へ「理想的な環境」を無理やり導入するのはコストを伴うので、無理をするのはよくない。

「バランスが肝心」

Master of Cloud/Mobile Engineer

Master of Cloud/Mobile Engineer

困難を克服しつつ、チャンスは利用したい

- **Server**

- **Google Cloud Platform**
 - PaaS: Google App Engine
- Slim3
 - Google App Engine専用のFramework
- **Maven**を使ってコンパイル、UnitTestを実行、レポート...

- **Client**

- **JavaScript**
 - Coffee Script, Type Script等
 - Less, Sass
 - 時代に合わせてFrameworkを選定
 - **Yo, Bower, Grunt**を使ってJavaScript, UnitTestを実行、CSSにコンパイル...
-

Master of Cloud/Mobile Engineer

プログラムを書けるだけのプログラマではなく、アプリケーションを構築できるプログラマを技術、姿勢の面で育成する

- Issue Tracker、ソースコード管理
 - Maven, Grunt等のビルドツール
 - Unit testツール
 - 上記を使ったCIの運用、自動化の重要性
 - PaaS上で稼働するアプリケーションの開発
 - ブラウザ上で稼働するアプリケーションの開発
 - 継続的な学習の重要性
-

Master of Cloud/Mobile Engineer

アプリケーション構築において、プログラミングは小さい要素。開発時や運用時の環境(**Platform**)をどれだけ活用できるか、開発～配備、分析までの運用をうまく回せるか、が重要。

- プログラミングは独学でも学べるし、言語レベルで良い悪い言うのはアプリケーション構築経験の浅い人
 - アプリケーションのユーザにはプログラミング言語がどうたらは関係ない
-

Platform - Google Cloud Platform

Google Cloud Platform



- Google App Engine
 - PaaS



- Google Compute Engine
 - IaaS



- Google Cloud Storage
 - SaaS



- Google Cloud SQL
 - DBaaS



- Google BigQuery
 - Analyze big data
- More
 - Prediction API, Translate API

<https://cloud.google.com/>

Google Cloud Platform

Google App Engine

- Platform as a Service
- Auto scale out
- Easy to
 - develop, deploy, maintain
 - Small start, Inexpensive
 - Not need to engineer for infrastructure so you can focus to build your application
- nice SDK
- Admin Console



<https://cloud.google.com/products/>

Google Cloud Platform

Google App Engine

- Enterprise
 - SulAmerica (2012, Brazil)
- Social
 - Applibot (2012, Japan)
- Campaign
 - Mixi (2009, Japan)
 - Tokyu Hands (2010, Japan)
 - Royal wedding (2011, United Kingdom)
- There are many cases since 2008
 - 2008/4 Python, 2009/4 Java, 2011/5 Golang
 - 2011/11 G/A



Google Cloud Platform

Google Compute Engine

- Infrastructure as a Service
 - similar to Amazon EC2
- Tools
 - Web Interface
 - Command line tools
 - API(OAuth2/https/json)



<https://cloud.google.com/products/compute-engine>

Google Cloud Platform

Google Cloud Storage

- Storage as a Service
 - similar to Amazon S3
- Tools
 - Web Interface
 - Command line tools
 - API(OAuth2/https/json)



<https://cloud.google.com/products/cloud-storage>

Google Cloud Platform

Google Cloud SQL

- Database as a Service
 - similar to Amazon RDS
- Tools
 - Web Interface
 - Command line tools
 - API(OAuth2/https/json)



<https://cloud.google.com/products/cloud-sql>

Google Cloud Platform

BigQuery

- *"Analyze Big Data in the cloud using SQL and get real-time business insights in seconds using Google BigQuery. Use a fully-managed data analysis service with no servers to install or maintain."*
- **Tools**
 - Web Interface
 - Command line tools
 - API(OAuth2/https/json)



<https://cloud.google.com/products/big-query>

Google Cloud Platform

BigQuery



Speed - Analyze billions of rows in seconds.

- 1 terabyte/second



Scale - Terabytes of data, trillions of records.



Simplicity - SQL-like query language, hosted on Google infrastructure.

But Not Index... **YES, TABLE SCAN !!**

Google Cloud Platform

Other Products

- Prediction API
 - *"Tap into Google's **machine learning** algorithms to analyze data and predict future outcomes."*
- Translate API
 - *"Use Google Translate API to build multilingual apps and **programmatically translate text** in your webpage or application."*

<https://cloud.google.com/products/more-products>

Google Cloud Platform

Recap

- Google provides several platform
 - IaaS: Google Compute Engine
 - PaaS: Google App Engine
 - S(torage)aaS: Google Cloud Storage
 - DBaaS: Google Cloud SQL
 - BigQuery is very unique platform
 - Analyzing big data: BigQuery
-

Google Cloud Platform

Recap

- Google provides several platform
 - IaaS: Google Compute Engine
 - PaaS: Google App Engine
 - S(torage)aaS: Google Cloud Storage
 - DBaaS: Google Cloud SQL
 - **DBaaS: Google Cloud Datastore**
 - **BaaS: Mobile backend starter**
 - <https://developers.google.com/cloud/samples/mbs/>
 - BigQuery is very unique platform
 - Analyzing big data: BigQuery
-

Any Questions ?

午後の部はハンズオンと
なります

Hands-on

最近のアプリケーション開発事情

Platform

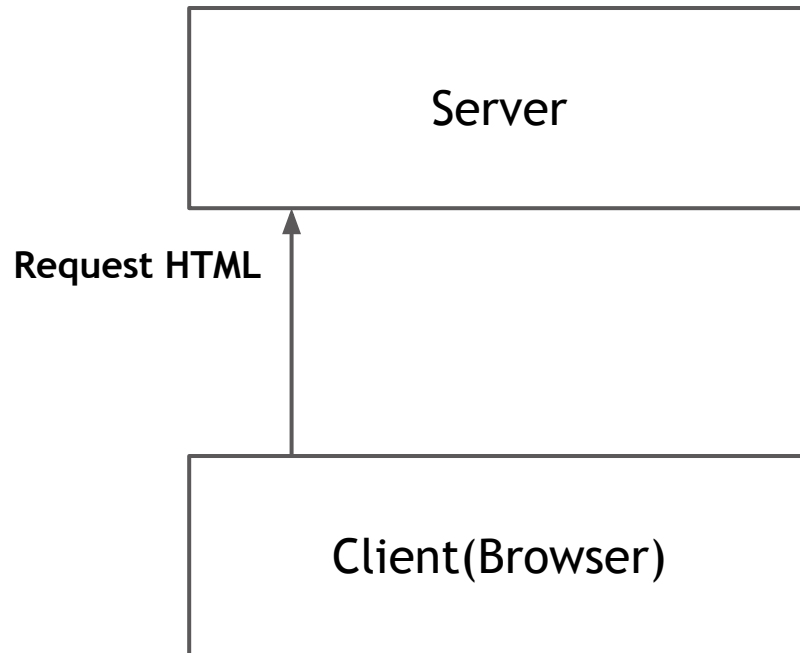
Server側はどんどんCloudに移行するのは間違いないので...

- Cloud(API) + Web Browser(JavaScript) on PC
- Cloud(API) + Web Browser(JavaScript) on Smart device
- Cloud(API) + Native(Java, iOS) on Smart device
- *BaaS(開発無) + Native(Java, iOS) on Smart device*

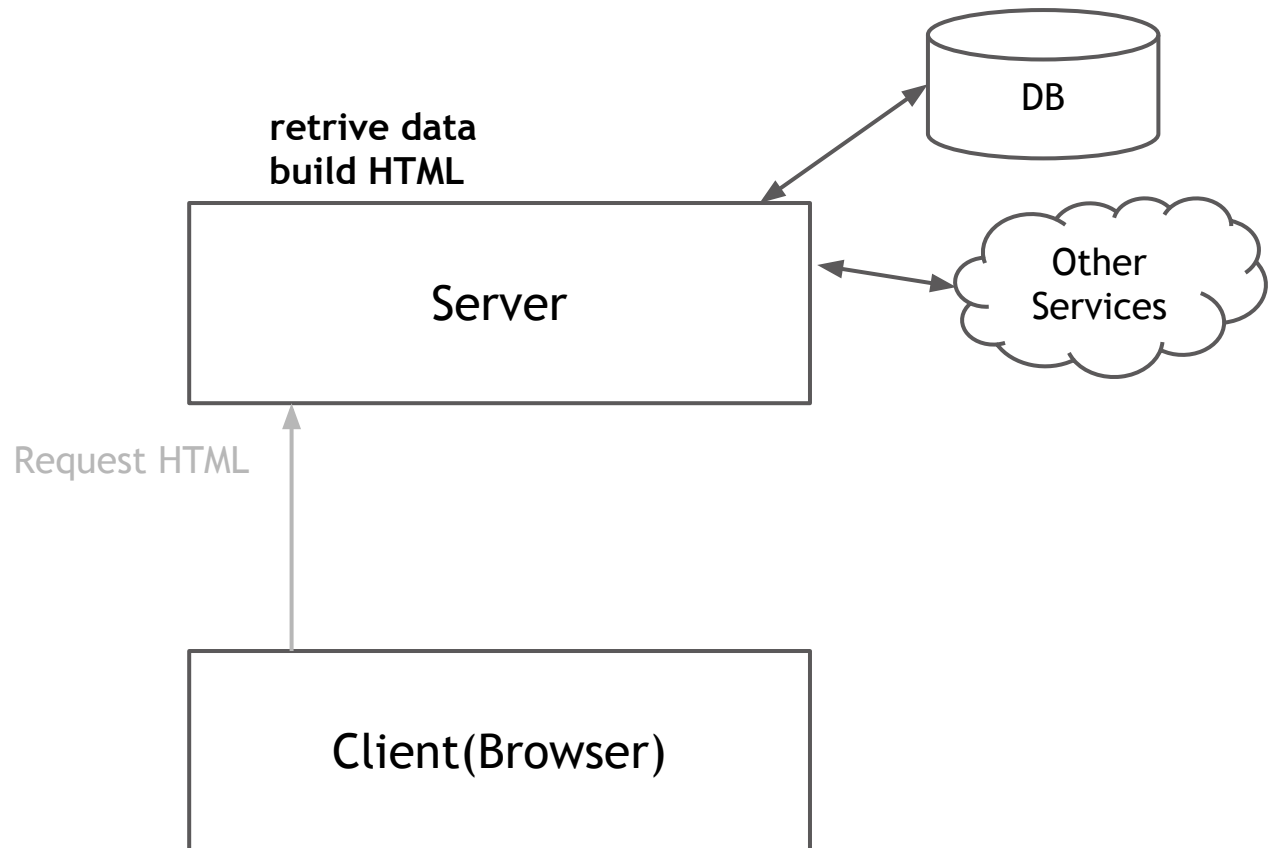
これらのパターンにはまるものが多くなる

Patterns - GAE Application

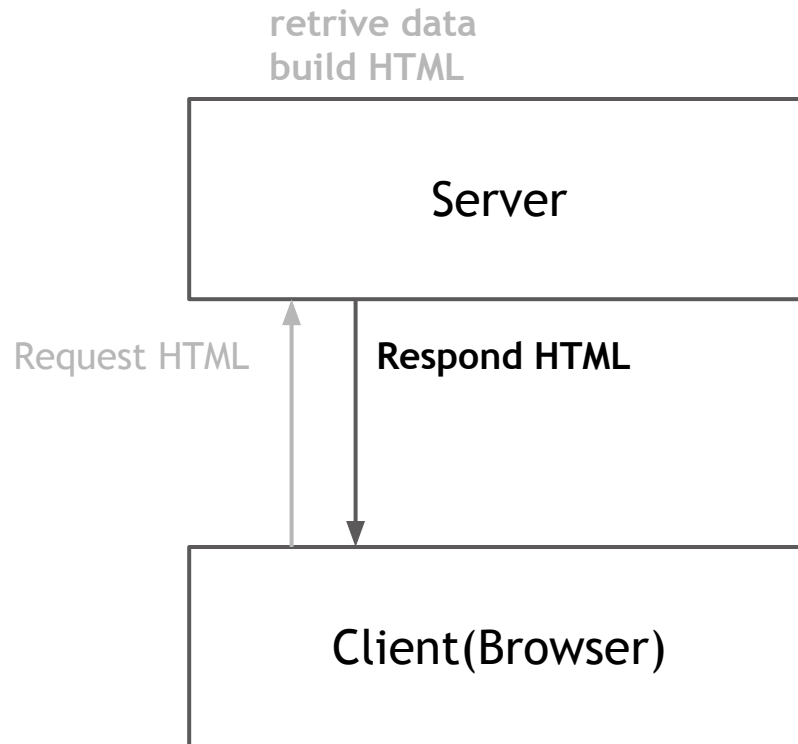
Sequence of old style Web App



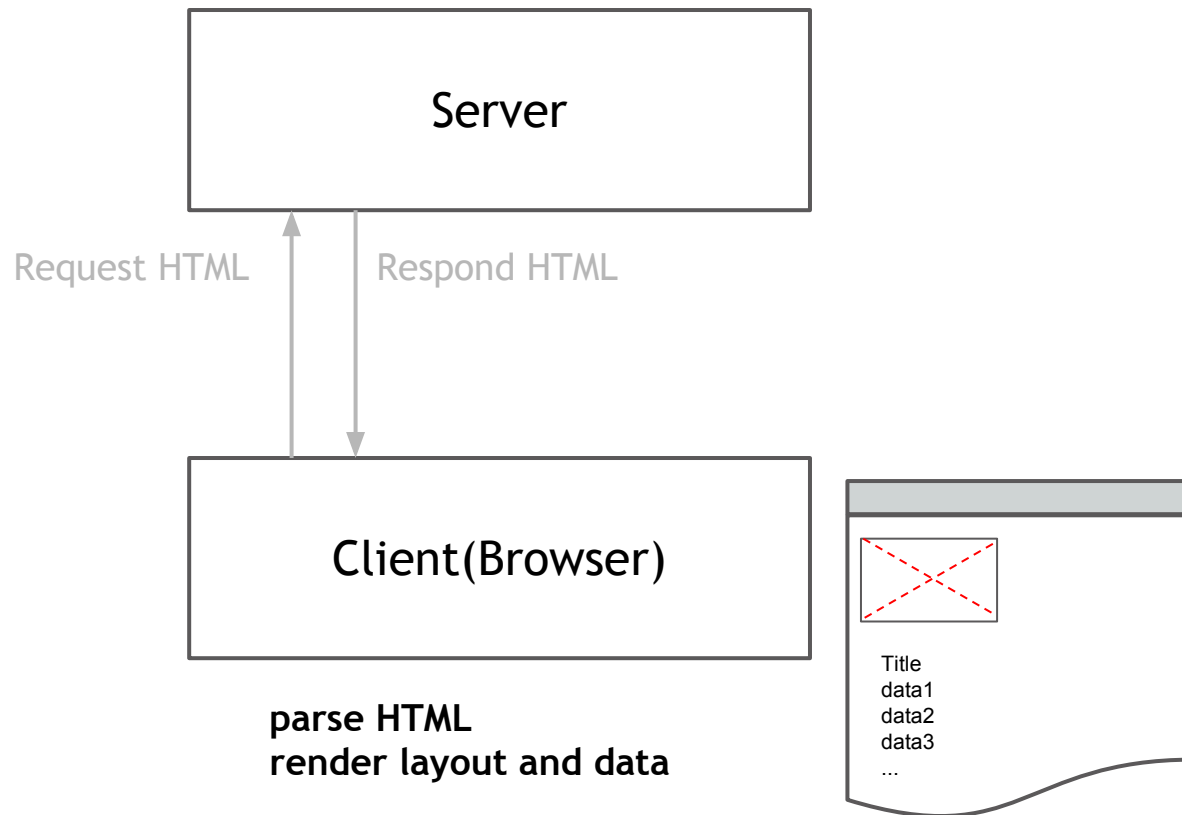
Sequence of old style Web App



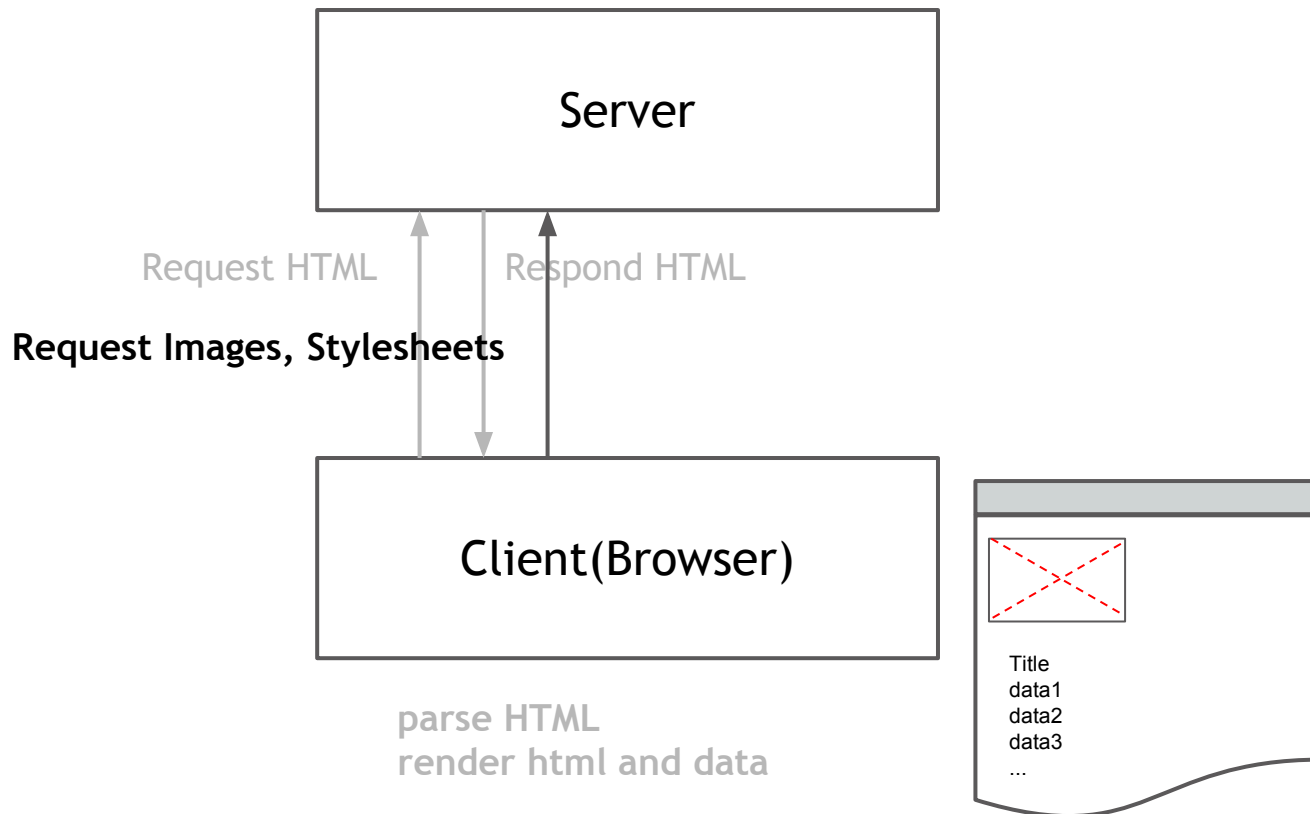
Sequence of old style Web App



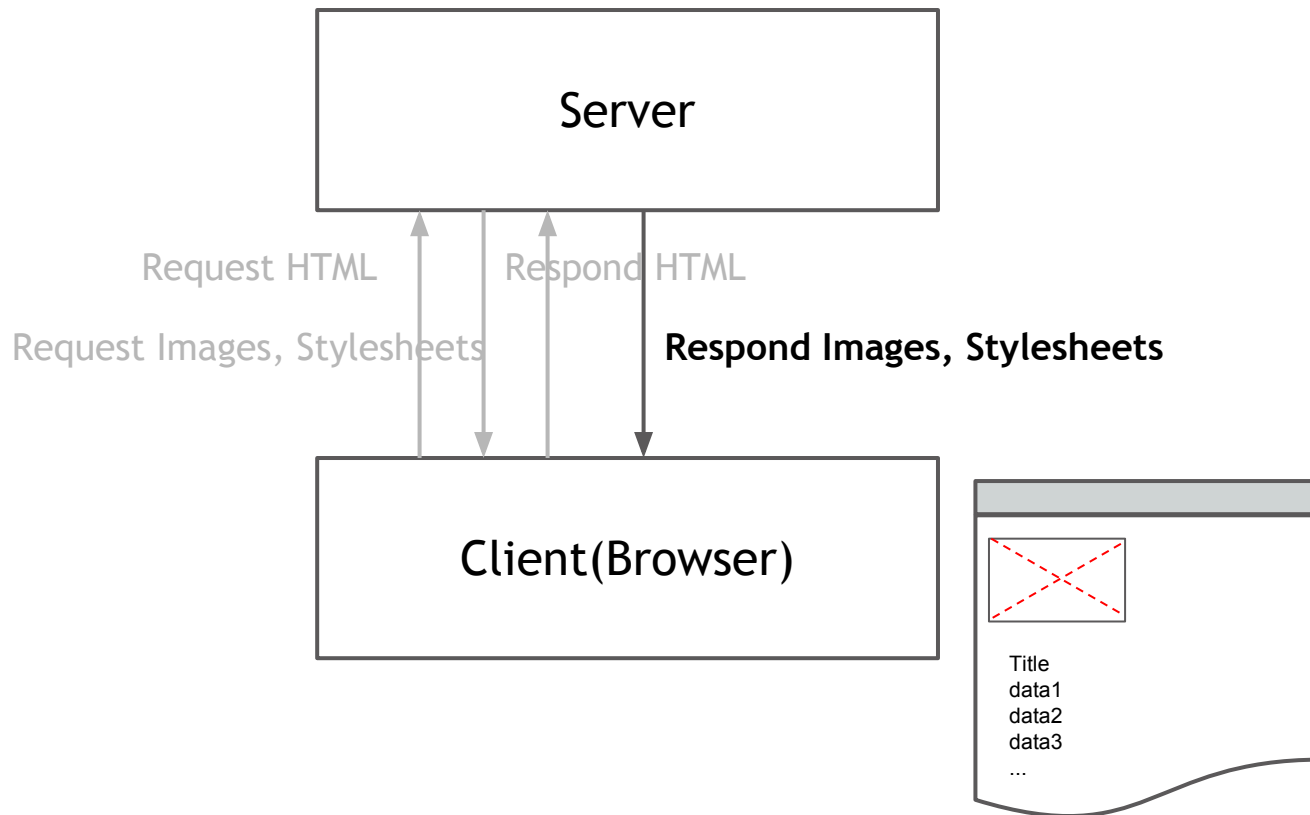
Sequence of old style Web App



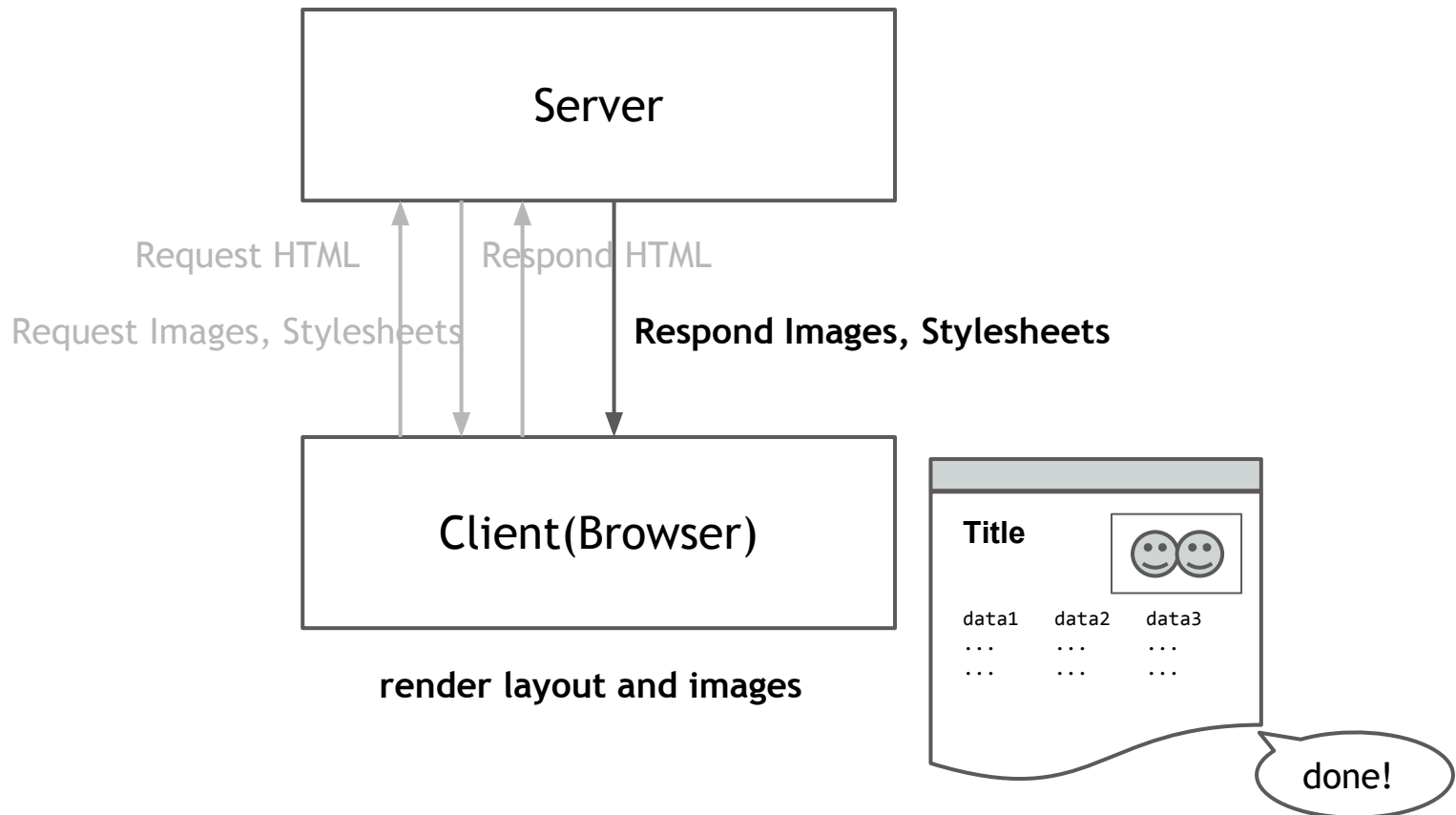
Sequence of old style Web App



Sequence of old style Web App



Sequence of old style Web App

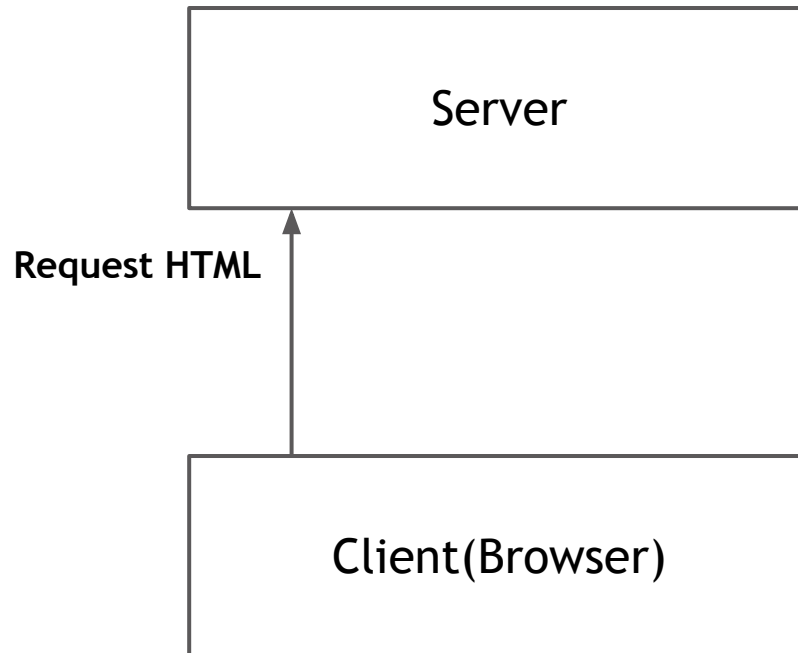


Sequence of old style Web App

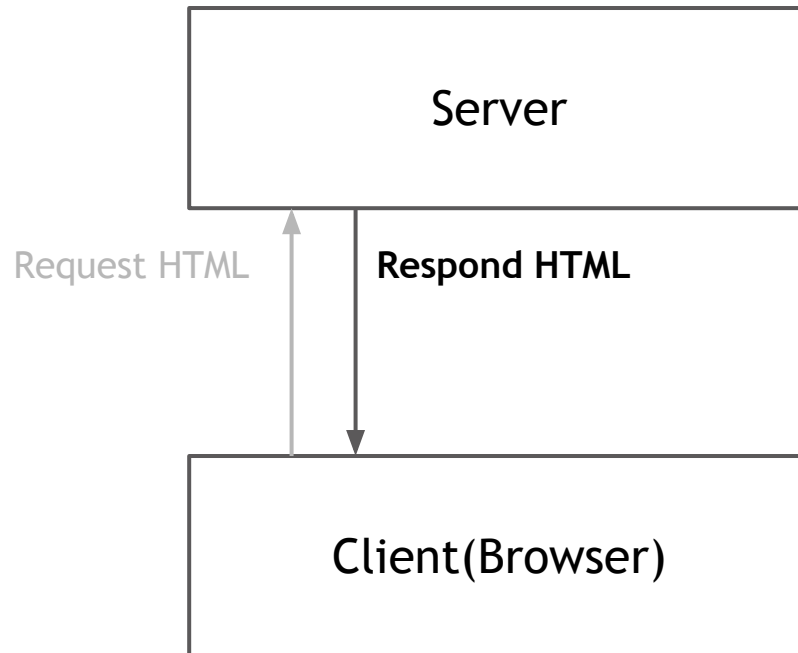
Recap

- Server respond
 - HTML with data (=dynamic content)
 - Usually use template engine to build html
 - JSP, JSF, ...
 - Other static resources
 - Images, Stylesheets, Javascripts,...
-

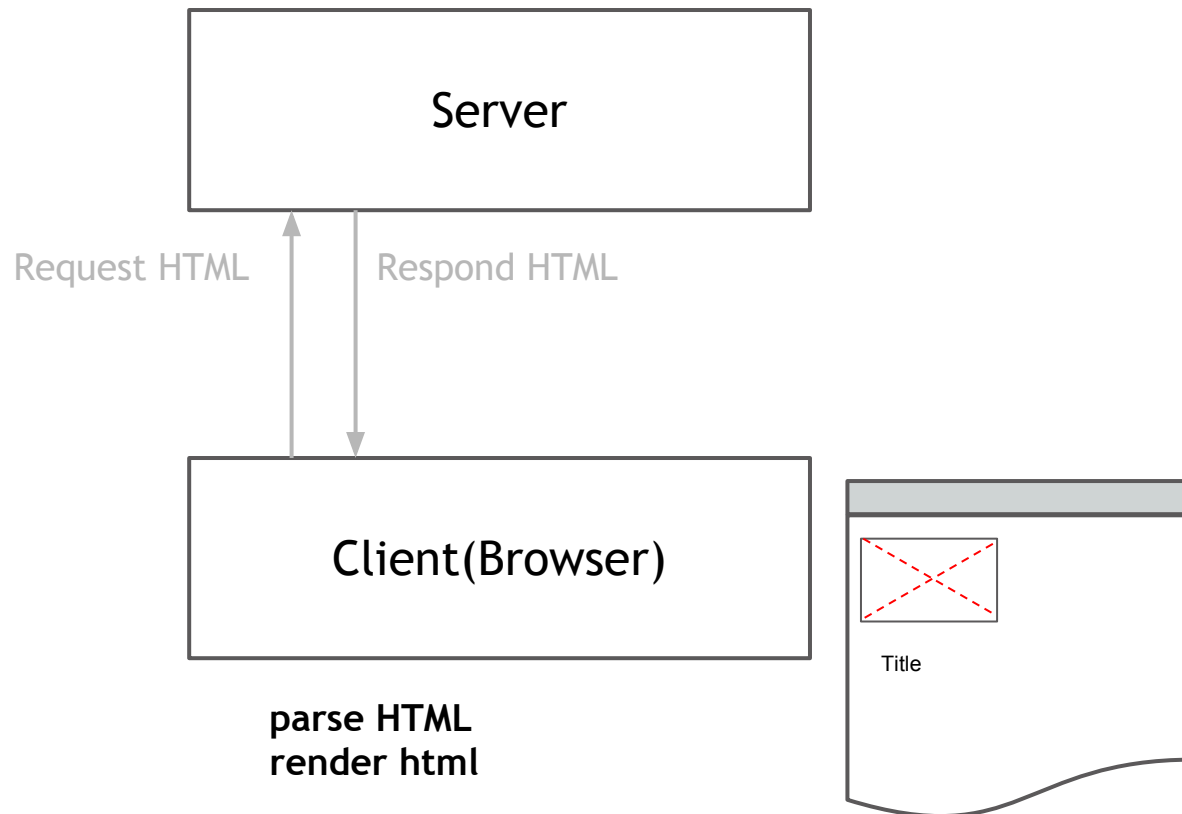
Sequence of new style Web App



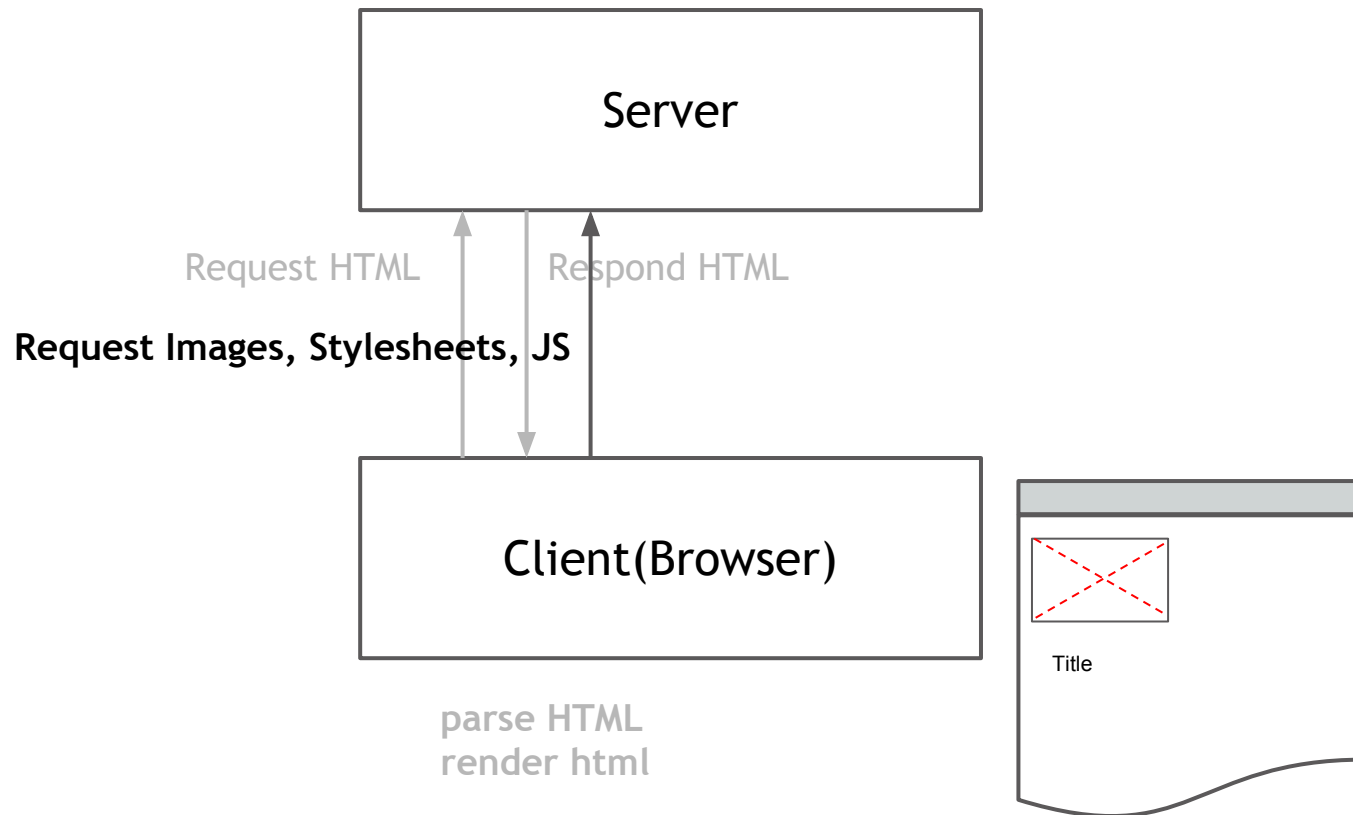
Sequence of new style Web App



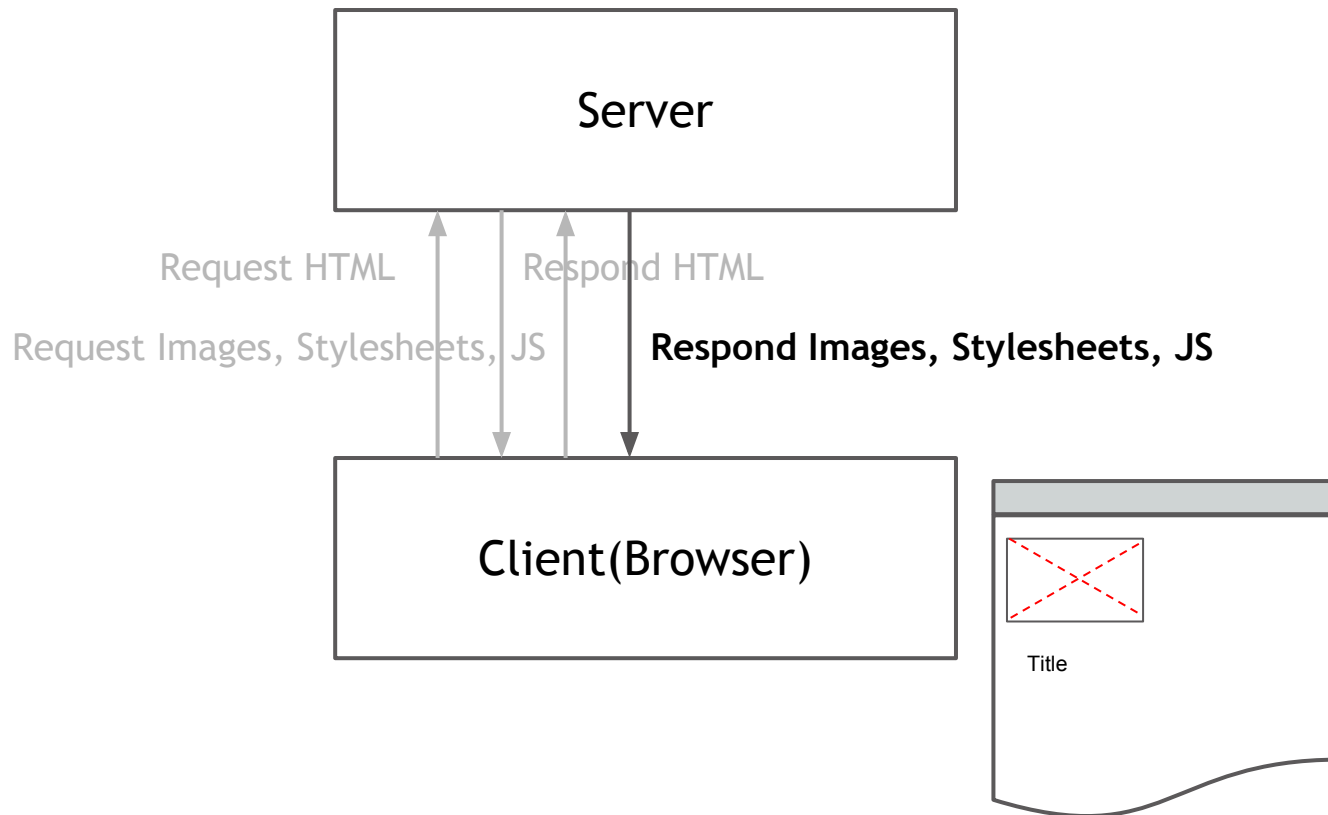
Sequence of new style Web App



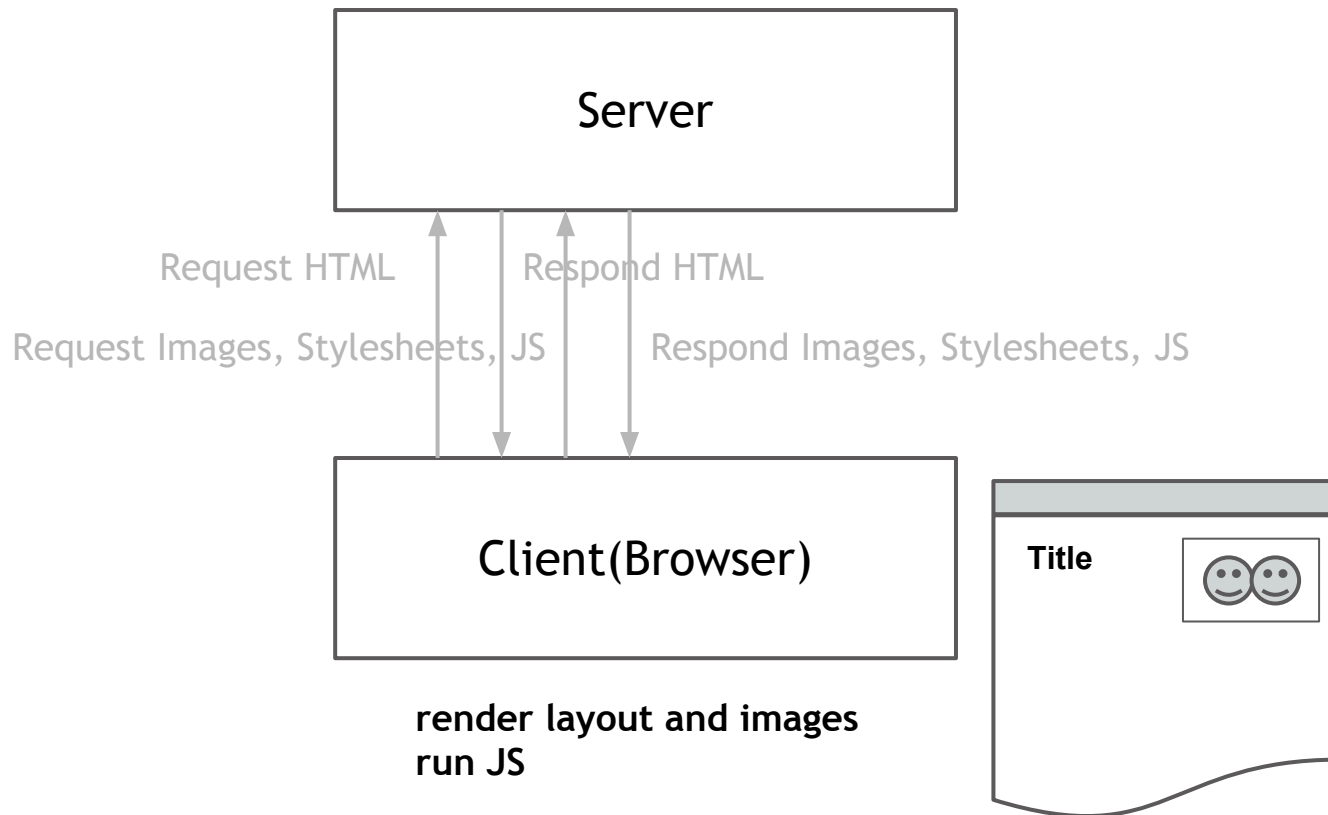
Sequence of new style Web App



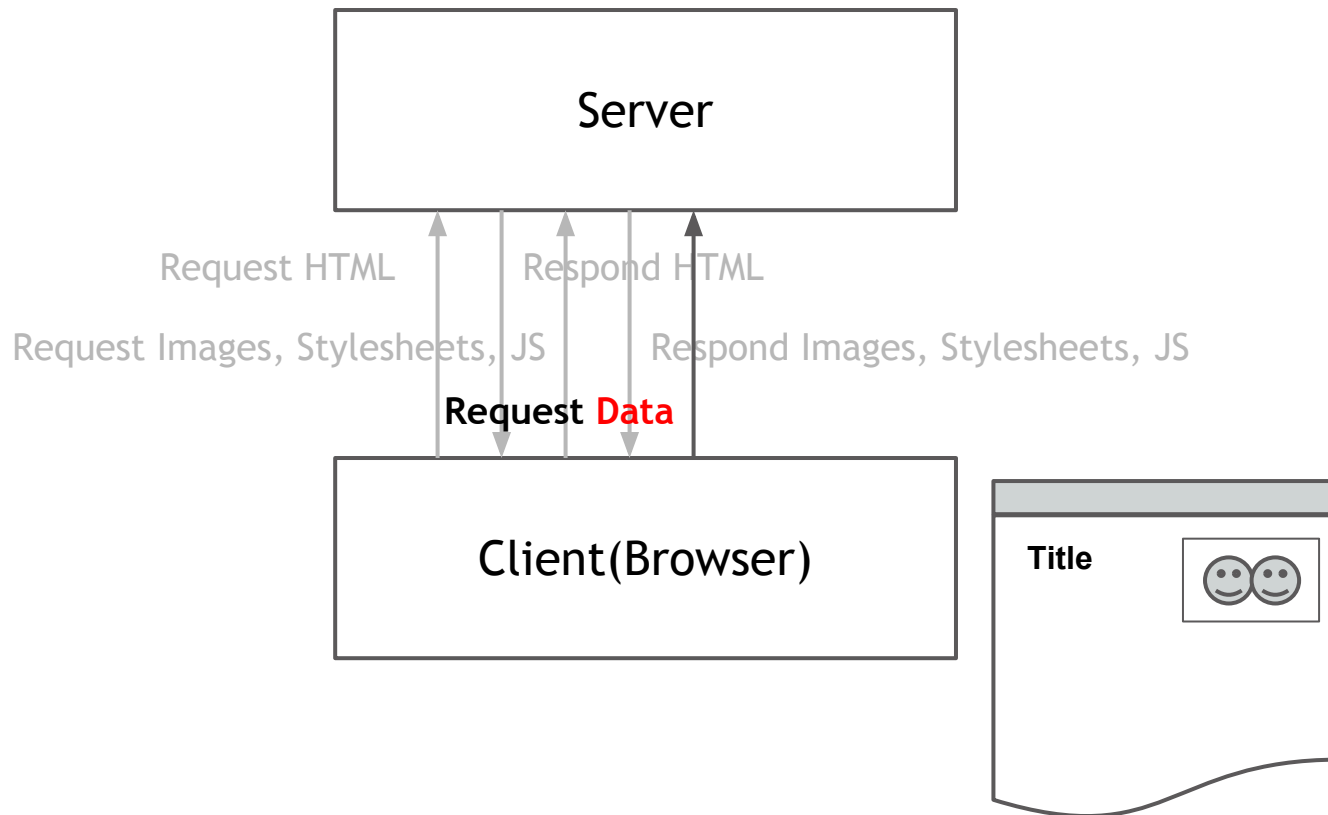
Sequence of new style Web App



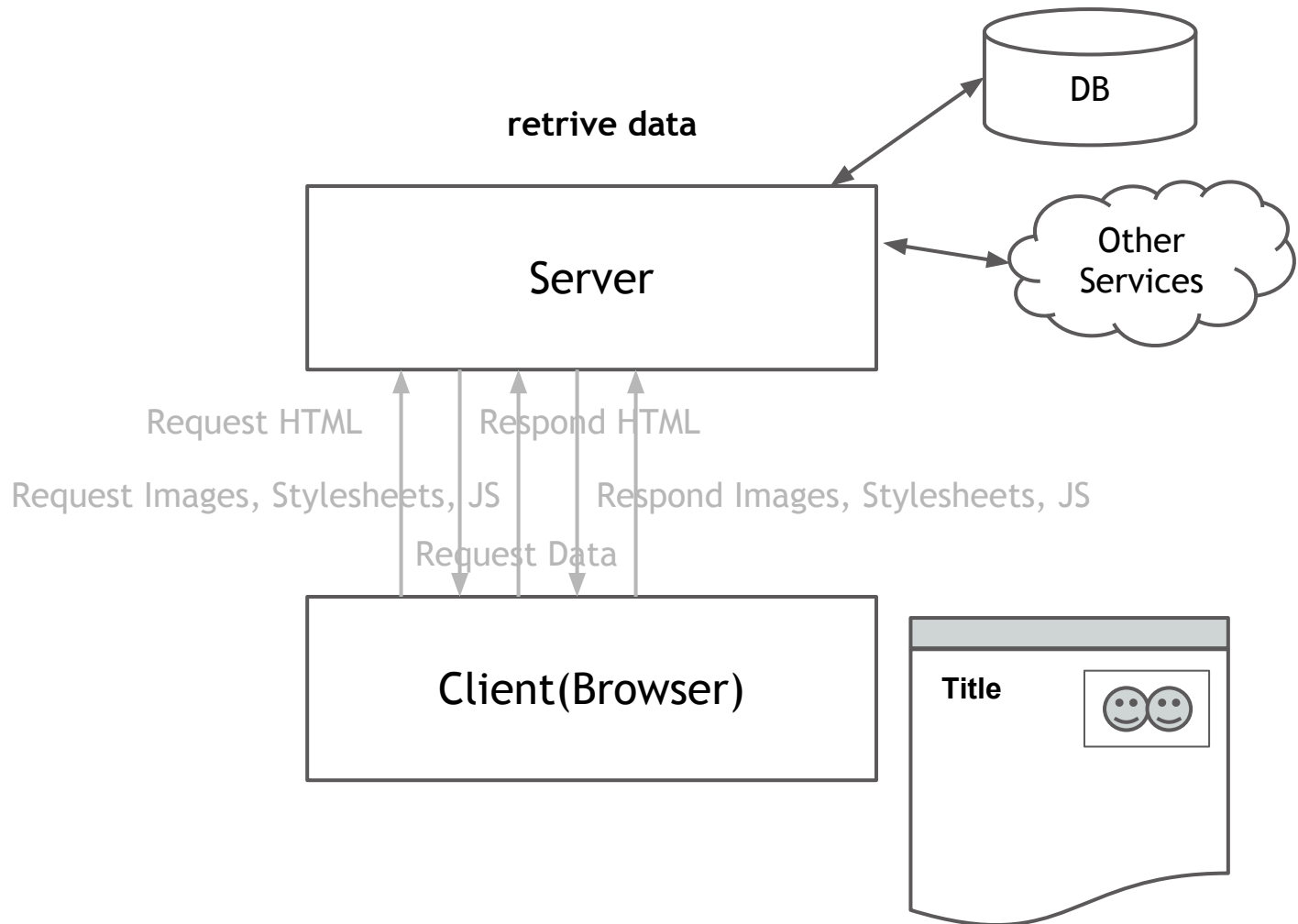
Sequence of new style Web App



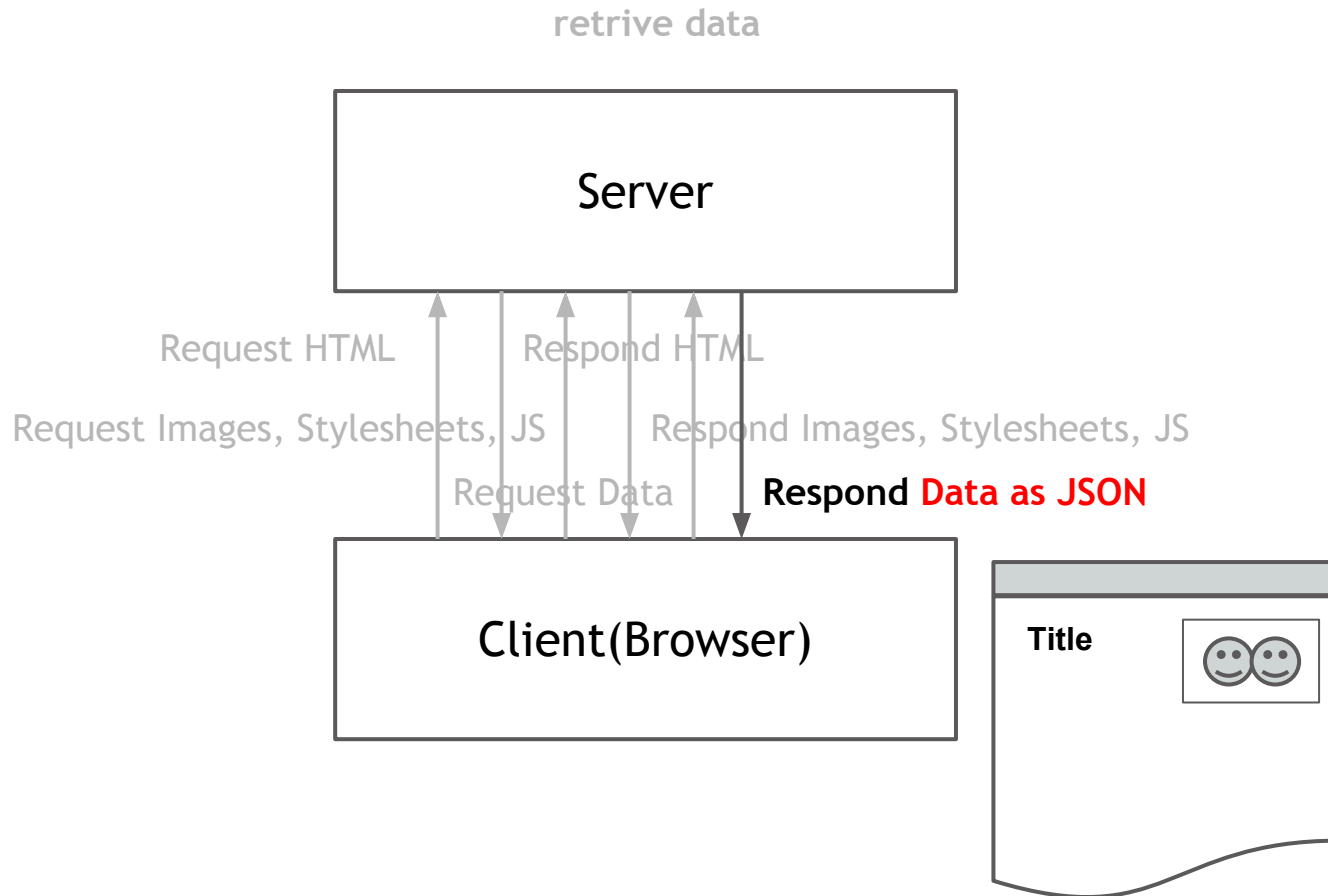
Sequence of new style Web App



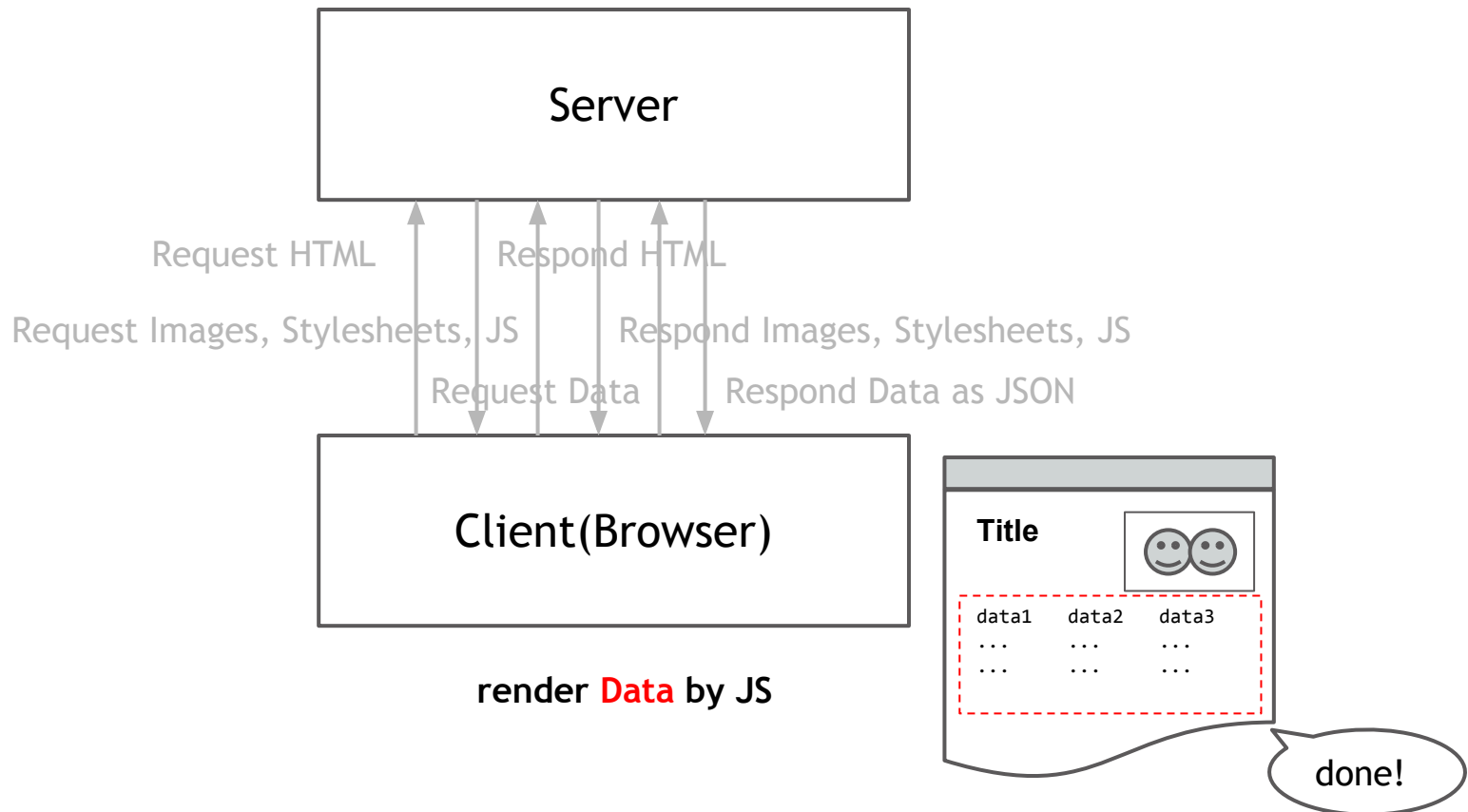
Sequence of new style Web App



Sequence of new style Web App



Sequence of new style Web App



Sequence of new style Web App

Recap

- Server respond
 - HTML
 - does not contain data
 - does not generate (=static content)
 - Other static resources
 - Images, Stylesheets, Javascripts,...
 - JSON(data)
 - Programs are run on both of server and client.
 - Server: Java, JavaScript, Python, PHP, Perl, Golang...
 - Client(Web browser): JavaScript
-

<http://goo.gl/8FRVSV>

TOP GATE
