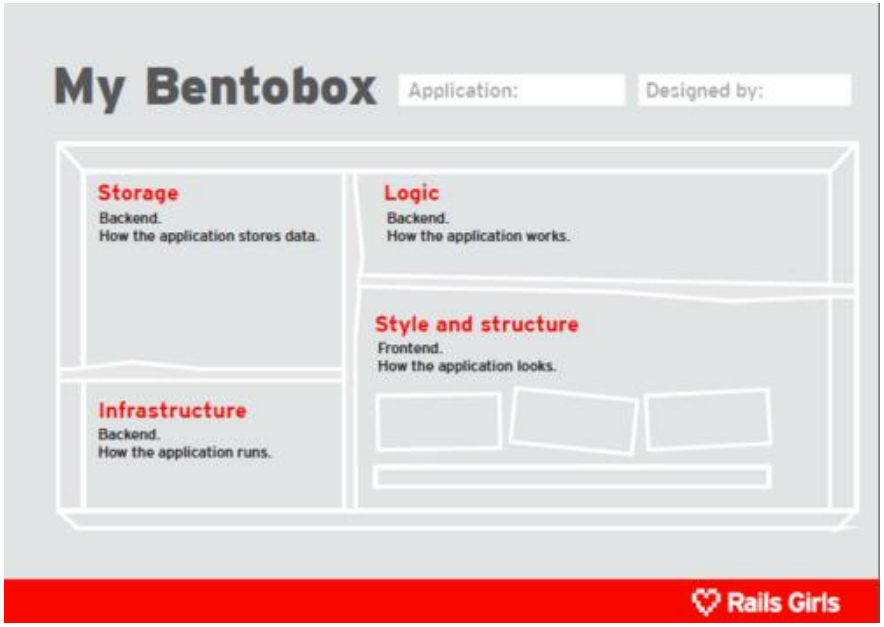
Breve introduzione allo sviluppo WEB

a cura di Ciro Attanasio - ciro.attanasio@email.cz

Rails Girls Udine

Partiamo (1 di 1)

Come funziona il WEB



e quali tecnologie lo compongono

Cos’è un Client (1 di 2)

Un client, in informatica, indica una componente che accede ai servizi o alle risorse di un'altra componente detta server.

Un computer collegato ad un server tramite una rete informatica (locale o geografica) ed al quale richiede uno o più servizi, utilizzando uno o più protocolli di rete, è un esempio di client hardware.

Cos’è un Client (2 di 2)

Cos’è un Server (1 di 2)

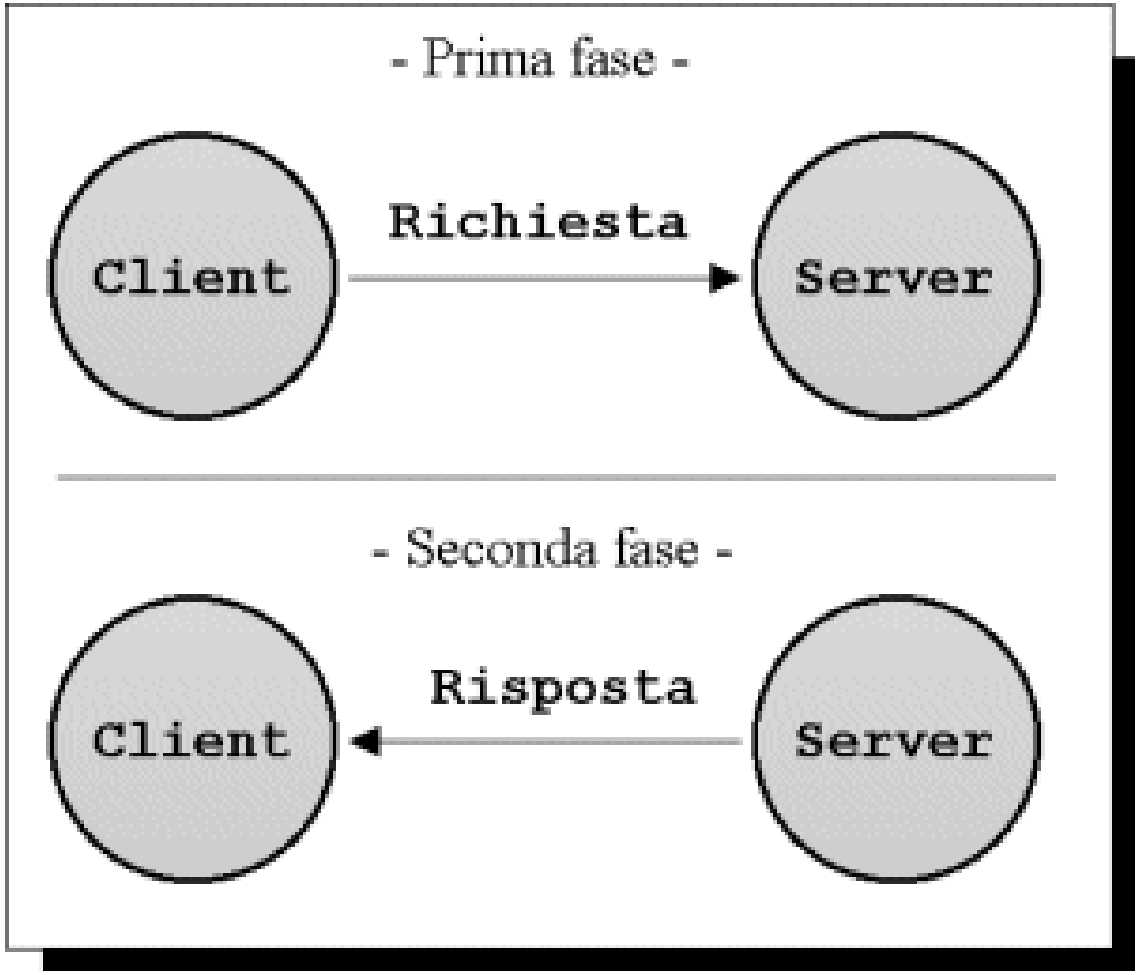
In informatica il termine server indica genericamente un componente o sottosistema informatico di elaborazione che fornisce, a livello logico e a livello fisico, un qualunque tipo di servizio ad altre componenti (tipicamente chiamate client, cioè "cliente") che ne fanno richiesta attraverso una rete di computer, all'interno di un sistema informatico o direttamente in locale su un computer.

Cos’è un Server (2 di 2)

Architettura Client-Server (1 di 4)

In informatica il termine sistema client-server indica un'architettura di rete nella quale genericamente un computer client si connette ad un server per la fruizione di un certo servizio, quale ad esempio la condivisione di una certa risorsa hardware/software con altri client, appoggiandosi alla sottostante architettura protocollare.

Architettura Client-Server (2 di 4)

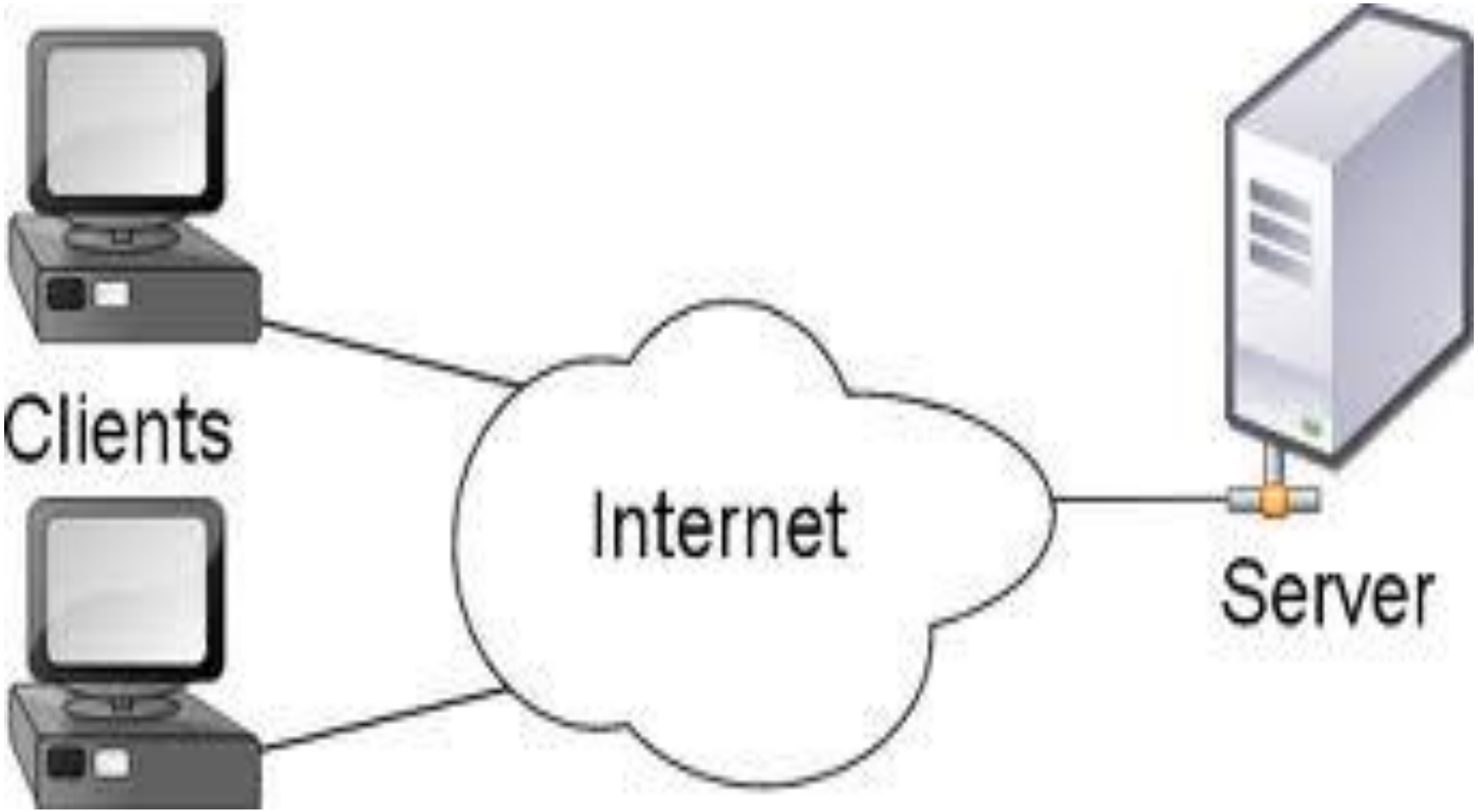


Architettura Client-Server (3 di 4)

Sono organizzati sotto forma di una tipica architettura client-server per la fruizione dei rispettivi servizi:

- Le reti locali aziendali (LAN);
- la rete Internet;
- i sistemi informatici;
- i sistemi operativi.

Architettura Client-Server (4 di 4)



Cos’è un Database (1 di 5)

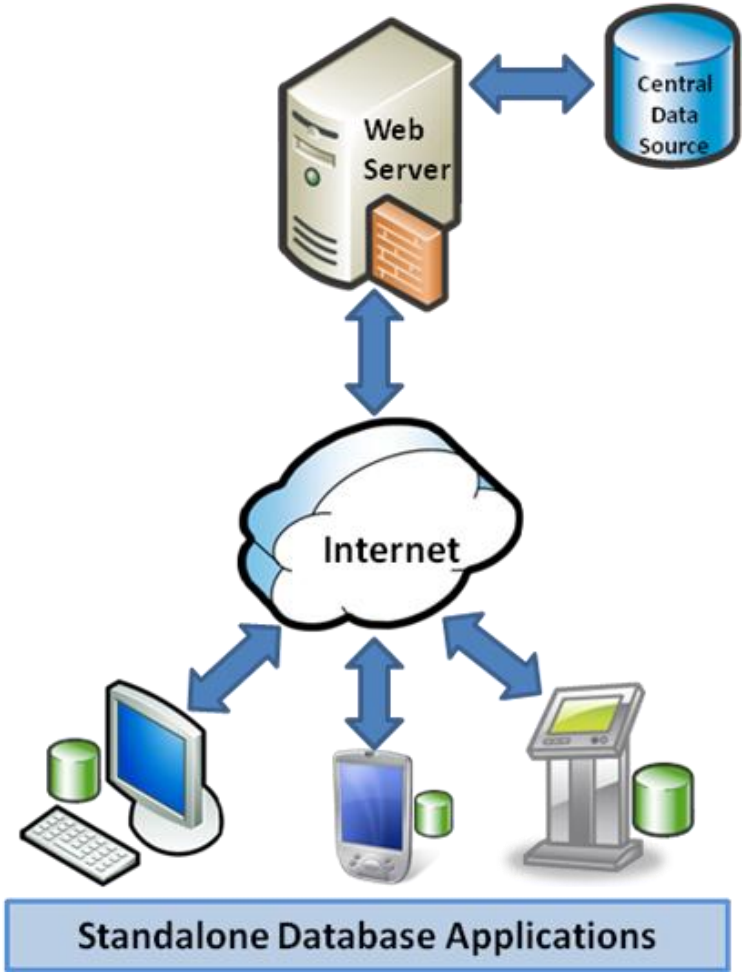
In informatica, il termine database, base di dati o banca dati (a volte abbreviato con la sigla DB), indica un archivio dati, o un insieme di archivi ben strutturati, in cui le informazioni in esso contenute sono strutturate e collegate tra loro secondo un particolare modello logico (relazionale, gerarchico, reticolare o a oggetti).

Cos’è un Database (2 di 5)

Cos’è un Database (3 di 5)

Sono collegate in modo tale da consentire la gestione/organizzazione efficiente dei dati stessi e l'interfacciamento con le richieste dell'utente attraverso i cosiddetti query language (query di ricerca o interrogazione, inserimento, cancellazione, aggiornamento ecc.) grazie a particolari applicazioni software dedicate (DBMS), basate su un'architettura di tipo client-server.

Cos’è un Database (4 di 5)



Cos’è un Database (5 di 5)

Tra i più diffusi DBMS open source troviamo:

- MySQL
- PostgreSQL

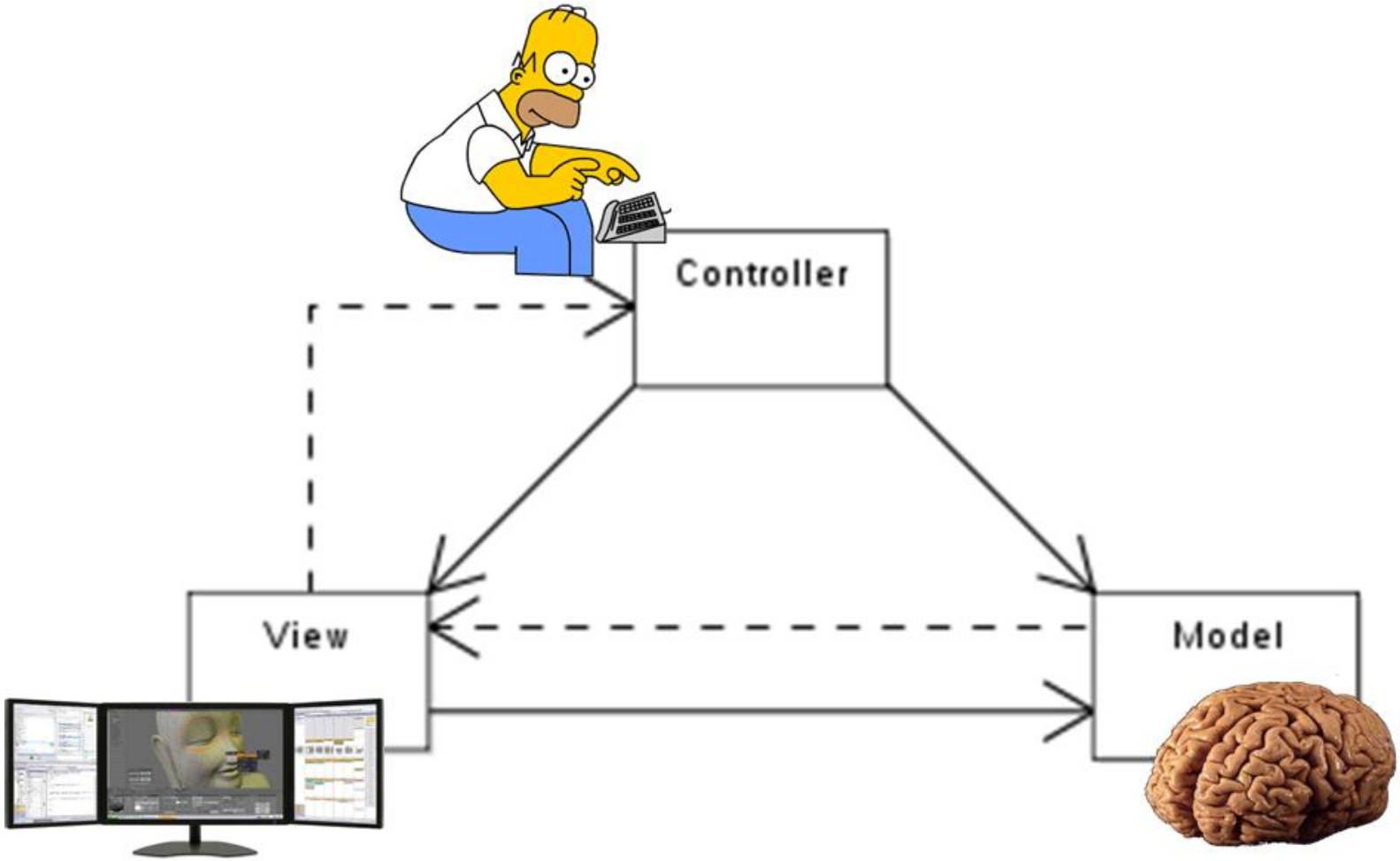
I più diffusi sistemi commerciali sono:

- Oracle
- IBM DB2
- Microsoft SQL Server
- Microsoft Access

Model-View-Controller (1 di 3)

Il Model-View-Controller, in informatica, è un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti, in grado di separare la logica di presentazione dei dati dalla logica di business.

Model-View-Controller (2 di 3)



Model-View-Controller (3 di 3)

Il funzionamento:

- Il browser (tramite il client) invia le richieste;
- Il controller interagisce con il model;
- Il controller chiama la view;
- La view produce la schermata sul browser Funzionamento.

Linguaggi lato Client (1 di 2)

Sono caratterizzate dal fatto che l'esecuzione e l'interpretazione delle istruzioni avvengono in locale sul computer che effettua la richiesta al server;

La conseguenza principale è che una pagina può essere visualizzata solo se gli utenti hanno a disposizione un software in grado di interpretare le istruzioni.

Linguaggi lato Client (2 di 2)

I principali linguaggi e tecnologie che possono essere utilizzati dal lato client sono:

- JavaScript
- Java
- VbScript
- Active X
- Flash
- Jquery

Linguaggi lato Server (1 di 2)

Le tecnologie dal lato server sono caratterizzate dal fatto che permettono l'esecuzione di script sul server, in modo quindi indipendente dall'ambiente di esecuzione dell'utente, tanto da impostazioni e preferenze, quanto dal browser, quanto dal sistema operativo.

Linguaggi lato Server (2 di 2)

Le principali e più comuni tecnologie dal lato server sono:

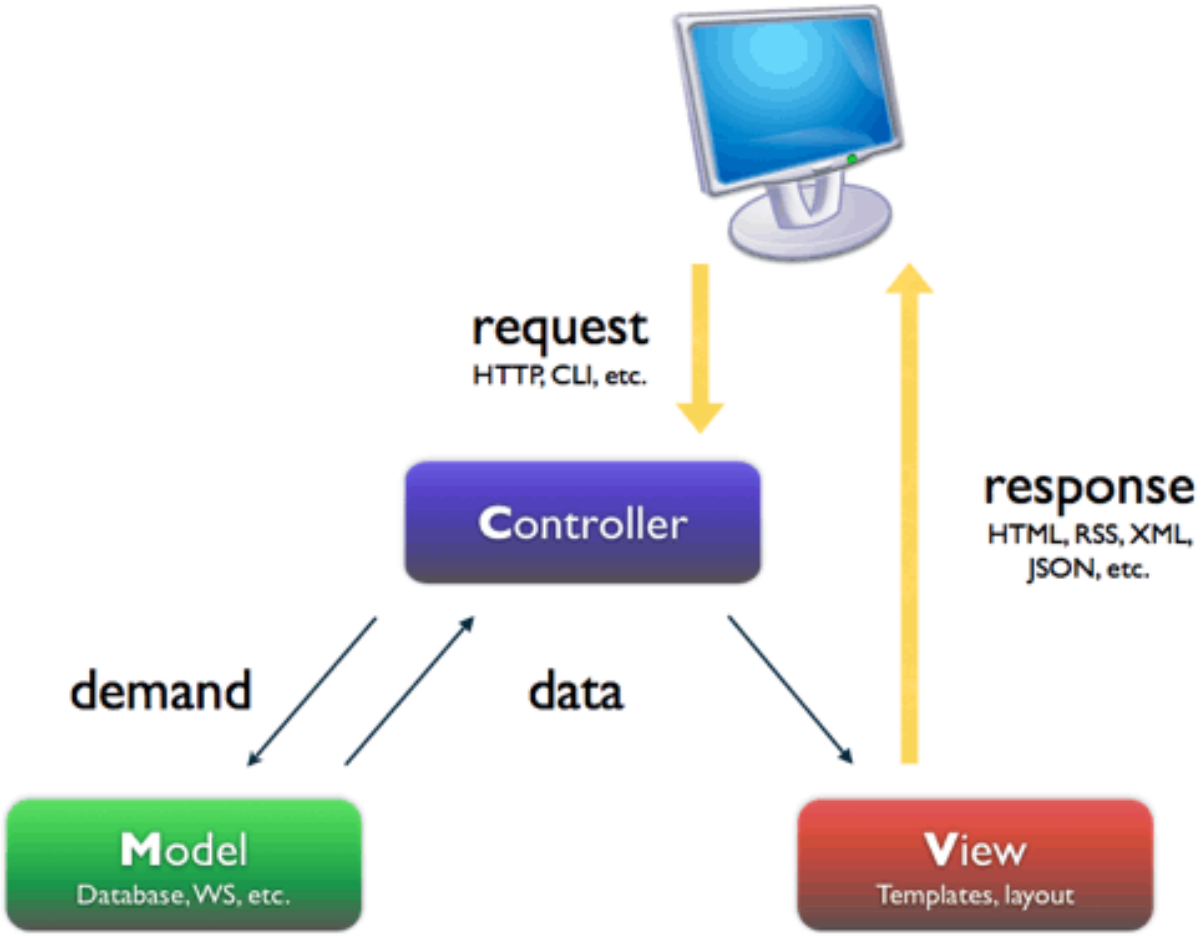
- Common Gateway Interface (CGI)
- Application Program Interface (API)
- Microsoft Active Server Pages (ASP)
- Java Server Pages (JSP)
- Cold Fusion Markup Language (CFML)
- PHP

Cos’è una Pagina Dinamica (1 di 3)

In una Pagina Dinamica i componenti vengono elaborati e composti solo nel momento in cui arriva una richiesta esplicita:

Viene utilizzata nei casi in cui sia necessario generare dei contenuti in modo automatico o in risposta ad un'operazione interattiva effettuata dall'utente;

Cos’è una Pagina Dinamica (2 di 3)



Cos’è una Pagina Dinamica (3 di 3)

Le tecnologie disponibili possono essere e divise in due gruppi principali:

- Linguaggi lato client;
- Linguaggi lato server.

La scelta dell'una o dell'altra avverrà in base a:

- Carico di lavoro da affidare a client e server;
- affidabilità del linguaggio.

Differenza HTML - HTML5 (1 di 3)

HTML 4 e HTML 5 sono entrambi due linguaggi markup;

Il 5 però non è, come immaginabile, il successore del 4 ma e destinato a diventare il nuovo standard;

Infatti, prima dell'HTML 5, lo standard proposto era l'XHTML (Extension HTML).

Differenza HTML - HTML5 (2 di 3)

HTML 4 è stato reso disponibile a natale del 1999 e prevedeva una prima reale adozione dei fogli di stile (CSS) per separare la struttura dallo stile.

Dopo il fallimento della seconda release di XHTML è iniziata la concezione di HTML 5.

Differenza HTML - HTML5 (3 di 3)

Quest'ultimo linguaggio si propone di migliorare la possibilità di strutturazione delle informazioni sul Web, oltre che accentuando la separazione tra stile, contenuti e struttura, anche tramite la creazione di nuovi tag come <video> e <audio> che meglio si adattano alla nuova evoluzione del WEB.

Cos’è un Framework (1 di 4)

In informatica, e specificatamente nello sviluppo software, un framework è una struttura logica di supporto (spesso un'implementazione logica di un particolare design pattern) su cui un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore.

Cos’è un Framework (2 di 4)

Alla base di un framework c'è sempre una serie di librerie di codice utilizzabili con uno o più linguaggi di programmazione, spesso corredate da una serie di strumenti di supporto allo sviluppo del software, come ad esempio un IDE, un debugger o altri strumenti ideati per aumentare la velocità di sviluppo del prodotto finito.

Cos’è un Framework (3 di 4)

L'utilizzo di un framework impone dunque al programmatore una precisa metodologia di sviluppo del software;

Quando si parla di Framework è doveroso ricordare il pattern MVC.

Cos’è un Framework (4 di 4)

I framework MVC, più diffusi, sono:

- Zend Framework (PHP)
- Ruby on Rails (Ruby)
- Symfony (PHP)
- Flex (Java)
- ASP.NET (ASP e C#)
- Yii (PHP)
- CakePHP (PHP)

Cos’è un Framework CSS (1 di 2)

Un framework è una struttura logica di supporto su cui un software può essere progettato e realizzato spesso facilitandone lo sviluppo;

Nel mondo del web design, per essere più specifici, un framework è l’insieme strutturato di file (HTML, CSS e Javascript), cartelle e codice standardizzato secondo una logica, che viene utilizzato come fondamenta o supporto per la creazione di un nuovo sito o applicazione web.

Cos’è un Framework CSS (2 di 2)

I framework CSS, più diffusi, sono:

- Bootstrap;
- Blueprint;
- 960 grid;
- YUI CSS;
- angularJS .

Bentobox (1 di 6)

Che cosa abbiamo imparato fino ad ora?

- La differenza tra client e server;
- La differenza tra un linguaggio lato cliente e un linguaggio lato server;
- La differenza tra un sito web e una applicazione web;
- La differenza tra un linguaggio di programmazione e un web framework;
- Regole di copia-incolla (Google everything ;-))

Bentobox (2 di 6)

Bentobox (3 di 6)

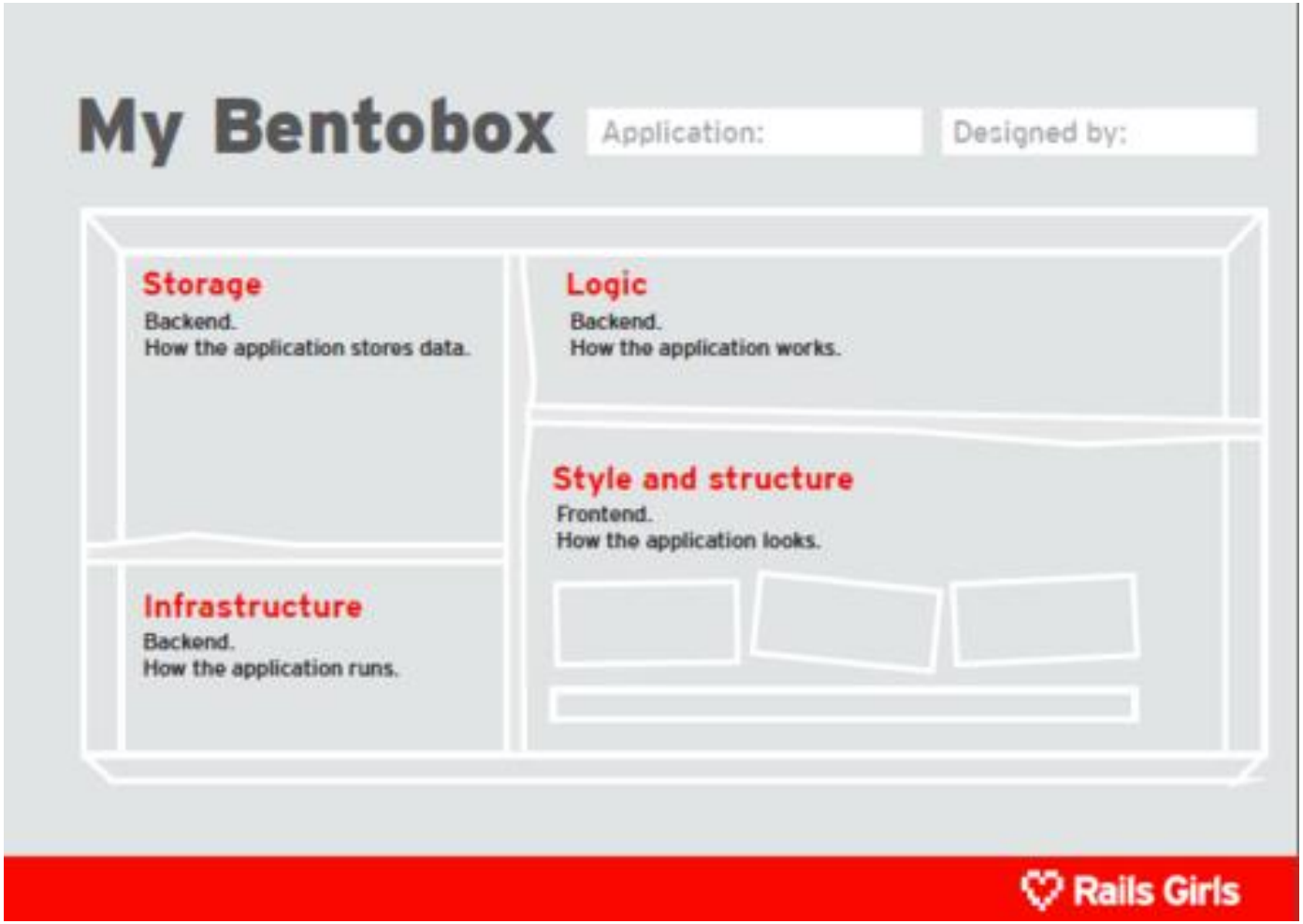
Perché Bento?

Il **bentō** (お弁当 *obentō*) è una sorta di vassoio contenitore con coperchio di varie forme e materiali contenente un pasto, in singola porzione, impacchettato in casa o comprato fuori, comune nella cucina giapponese;

il cui contenuto è disposto nel modo più efficiente possibile e in maniera graziosa;

È a tutti gli effetti un puzzle ;-)

Bentobox (4 di 6)

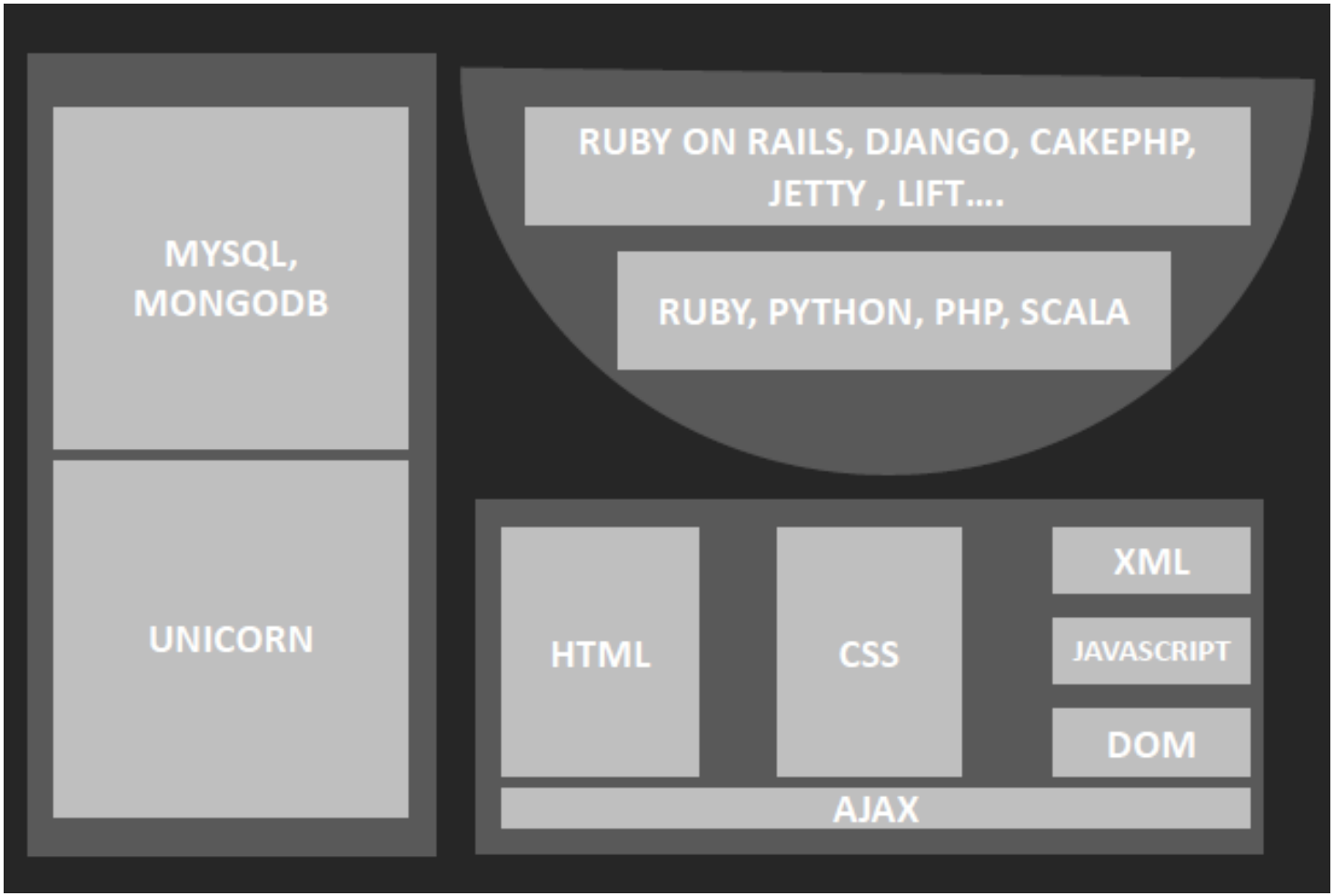


Bentobox (5 di 6)

Ora tocca a voi comporre la vostra Bento Box:

- Storage
- Infra
- The Logic
- The Structure e Style

Bentobox (6 di 6)



Cos’è Ruby on Rails (1 di 2)

Ruby on Rails (spesso chiamato RoR o semplicemente Rails) è un framework open source per applicazioni web scritto in Ruby da David Heinemeier Hansson per conto della 37signals;

la sua architettura è fortemente ispirata al paradigma Model-View-Controller (MVC).

Cos’è Ruby on Rails (2 di 2)

Ruby on Rails è :

stato creata nel 2005 e da allora ha acquisito una trazione globale impressionante;

una piattaforma open-source di sviluppo web, realizzata con il linguaggio di programmazione Ruby;

stato sviluppato implementando con particolare attenzione le funzionalità destinate all'interazione del framework con le basi di dati e le informazioni in esse archiviate.

A cosa serve Ruby on Rails (1 di 2)

Serve a sviluppare applicazioni in modo semplice diminuendo la percentuale di codice che solitamente va a ripetersi nelle applicazioni.

N.B.: Una grande contributo è dato dal pattern Model View Controller che è una pratica di programmazione che semplifica la separazione tra presentazione dei dati, logica della app e contenuti.

A cosa serve Ruby on Rails (2 di 2)

Il risultato è che le app sviluppate in Ruby on Rails sono particolarmente indicate per progetti dinamici, flessibili che necessitano aggiornamenti continui o ampliamenti futuri.

Framework CSS utilizzati con Rails

È possibile, per RoR, fare riferimento a :

- Bootstrap;
- Blueprint;
- 960 grid;
- YUI CSS.

Database utilizzati con Rails (1 di 2)

Tra i numerosi DBMS supportati da RoR, per citare soltanto alcuni di quelli rilasciati sotto licenza Open Source, è possibile fare riferimento a :

- MySQL;
- SQLite;
- PostgreSQL;
- MongoDB.

Database utilizzati con Rails (2 di 2)

database.yml

```
1  # SQLite version 3.x
2  #   gem install sqlite3-ruby (not necessary on OS X Leopard)
3  development:
4    adapter: sqlite3
5    database: db/development.sqlite3
6    timeout: 5000
7
8  # Warning: The database defined as 'test' will be erased and
9  # re-generated from your development database when you run 'rake'.
10 # Do not set this db to the same as development or production.
11 test:
12   adapter: sqlite3
13   database: db/test.sqlite3
14   timeout: 5000
15
16 production:
17   adapter: sqlite3
18   database: db/production.sqlite3
19   timeout: 5000
20
```

Model-View-Controller (1 di 5)

Le applicazioni sviluppate con Rails hanno una peculiarità, ovvero sono tutte organizzate secondo una struttura comune;

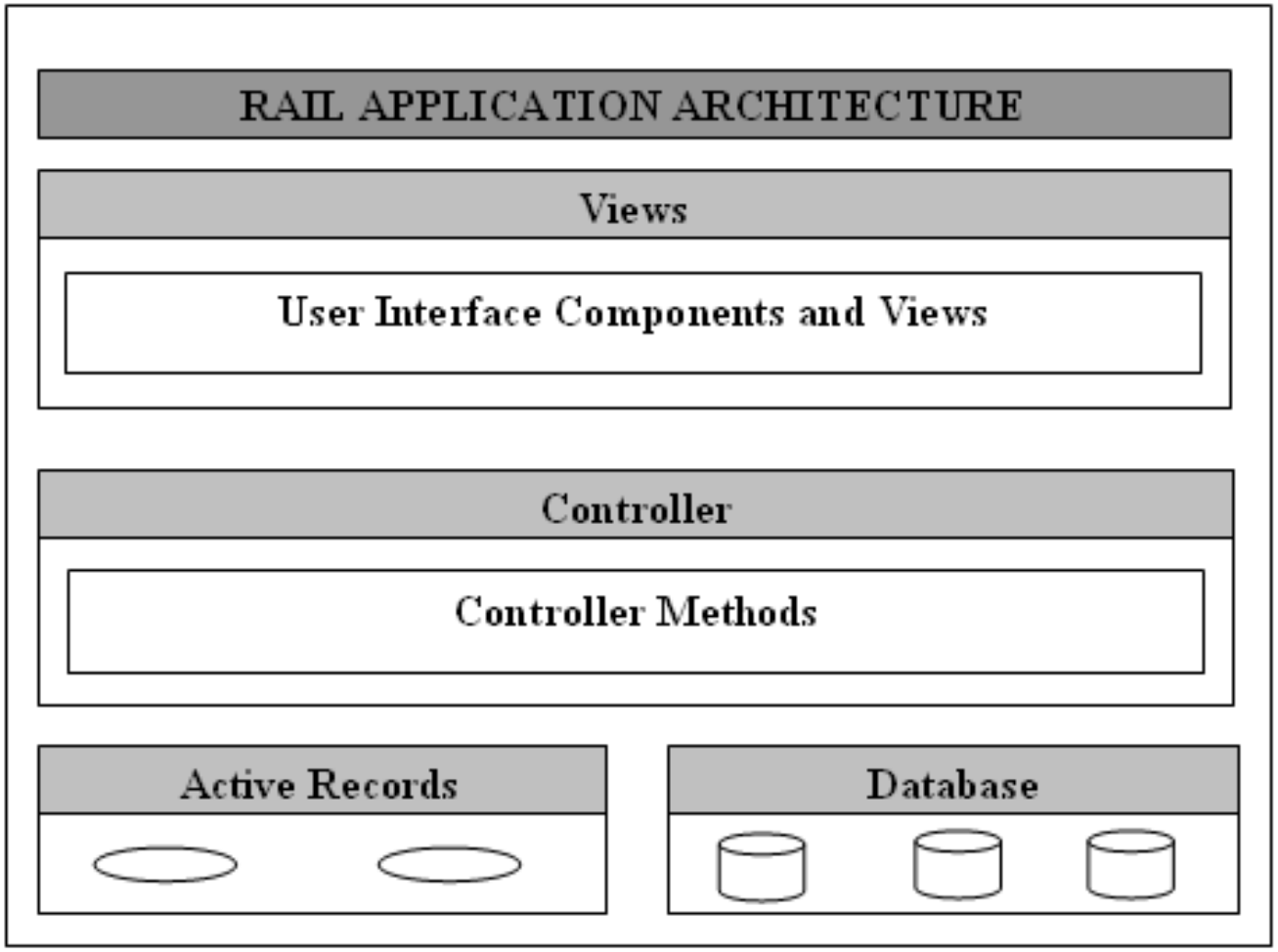
Questa è una conseguenza del fatto che il comando rails genera una serie di directory e file che forniscono una certa linea guida nello sviluppo, linea che se rispettata permette a Rails di effettuare molte cose automaticamente.

Model-View-Controller (2 di 5)

Esempio : caricare i file, generarli ed individuarli a runtime e molto altro;

Questa struttura comune permette anche di comprendere con semplicità il codice di progetti realizzati da altri, in quanto sono organizzati nella stessa maniera.

Model-View-Controller (3 di 5)

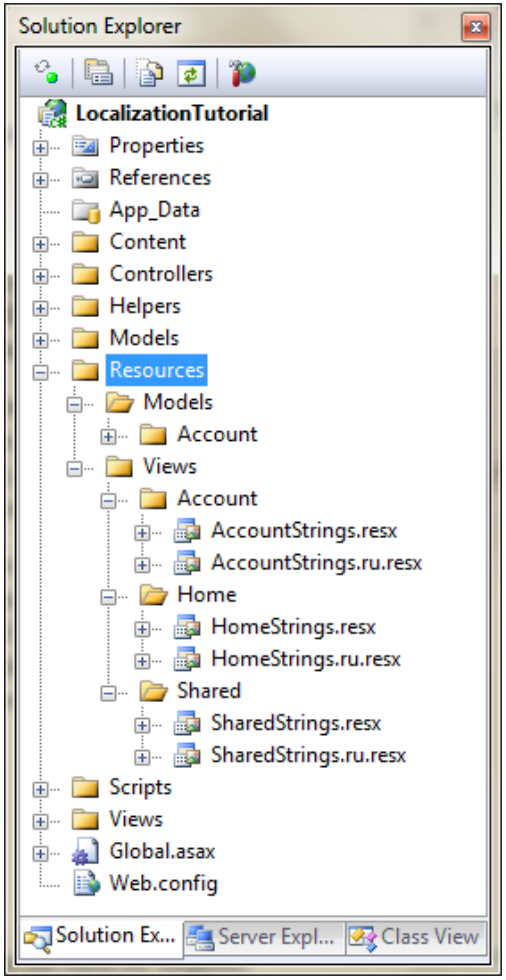


Model-View-Controller (4 di 5)

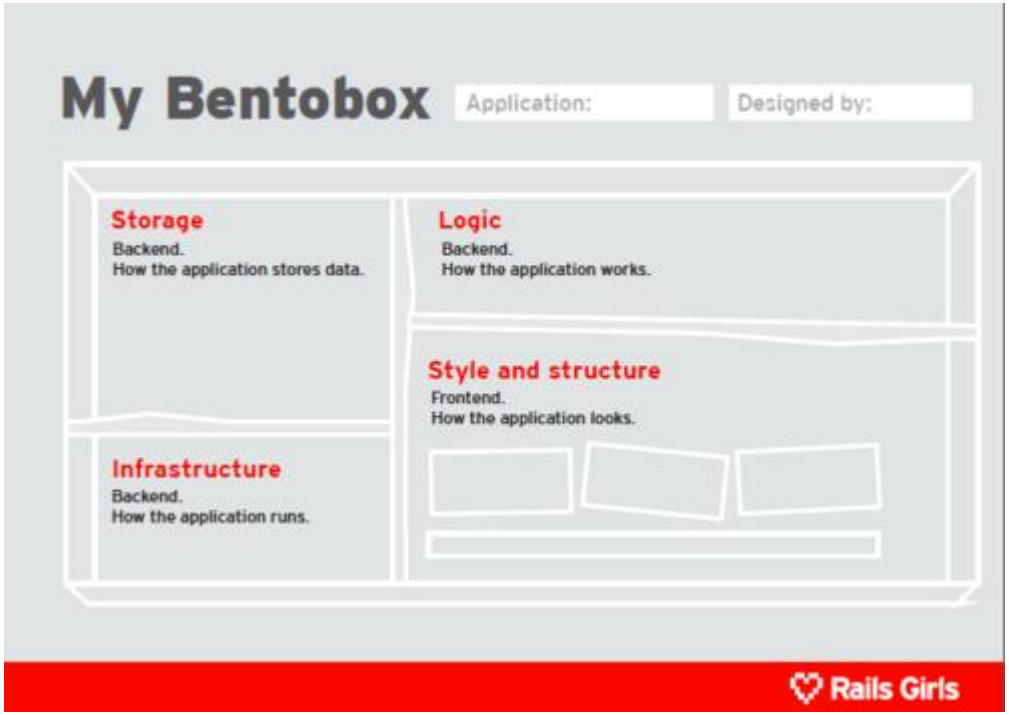
I dati (model) sono separati dall'interfaccia utente (view):

- Model
  - Accesso ai dati e alla logica di business
  - Indipendente dalla View e dal Controller
- Controller
  - Coordina l'interazione tra l'utente, le View, e il Model
- View
  - Presentazione dei dati e interazione con l'utente
  - Accesso in sola lettura al Model

Model-View-Controller (5 di 5)



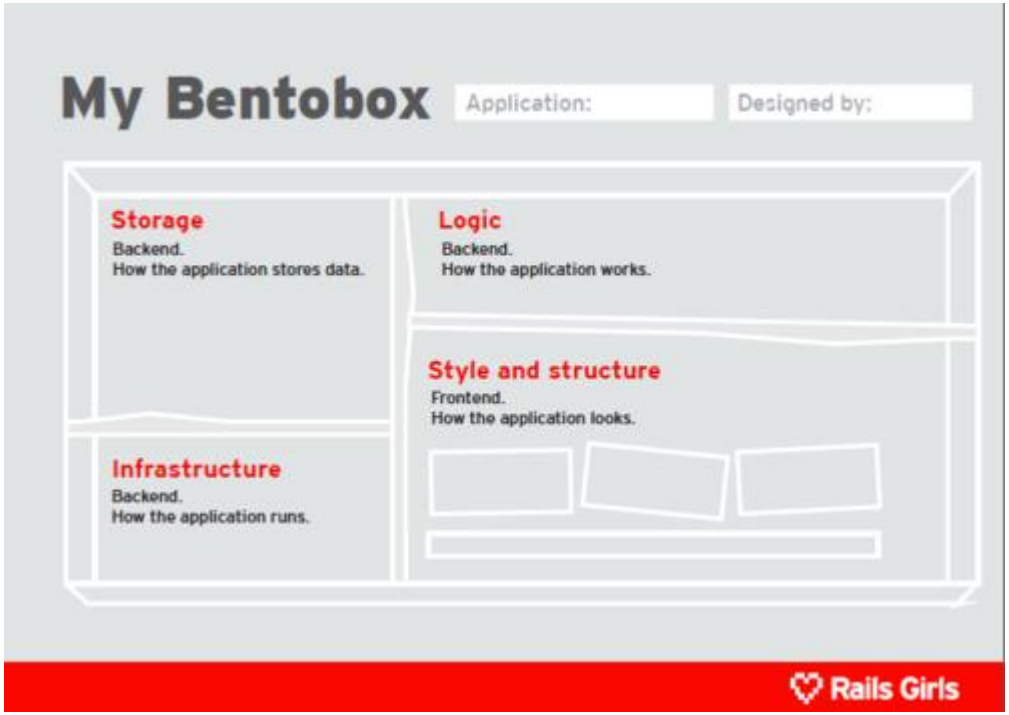
Fine



Grazie per l’attenzione

Breve introduzione allo sviluppo WEB

a cura di Ciro Attanasio - ciro.attanasio@email.cz



Come funziona il WEB e quali tecnologie lo compongono

Cos’è un Client (1 di 2)

Un client, in informatica, indica una componente che accede ai servizi o alle risorse di un'altra componente detta server.

Un computer collegato ad un server tramite una rete informatica (locale o geografica) ed al quale richiede uno o più servizi, utilizzando uno o più protocolli di rete, è un esempio di client hardware.

Cos’è un Client (2 di 2)

Cos’è un Server (1 di 2)

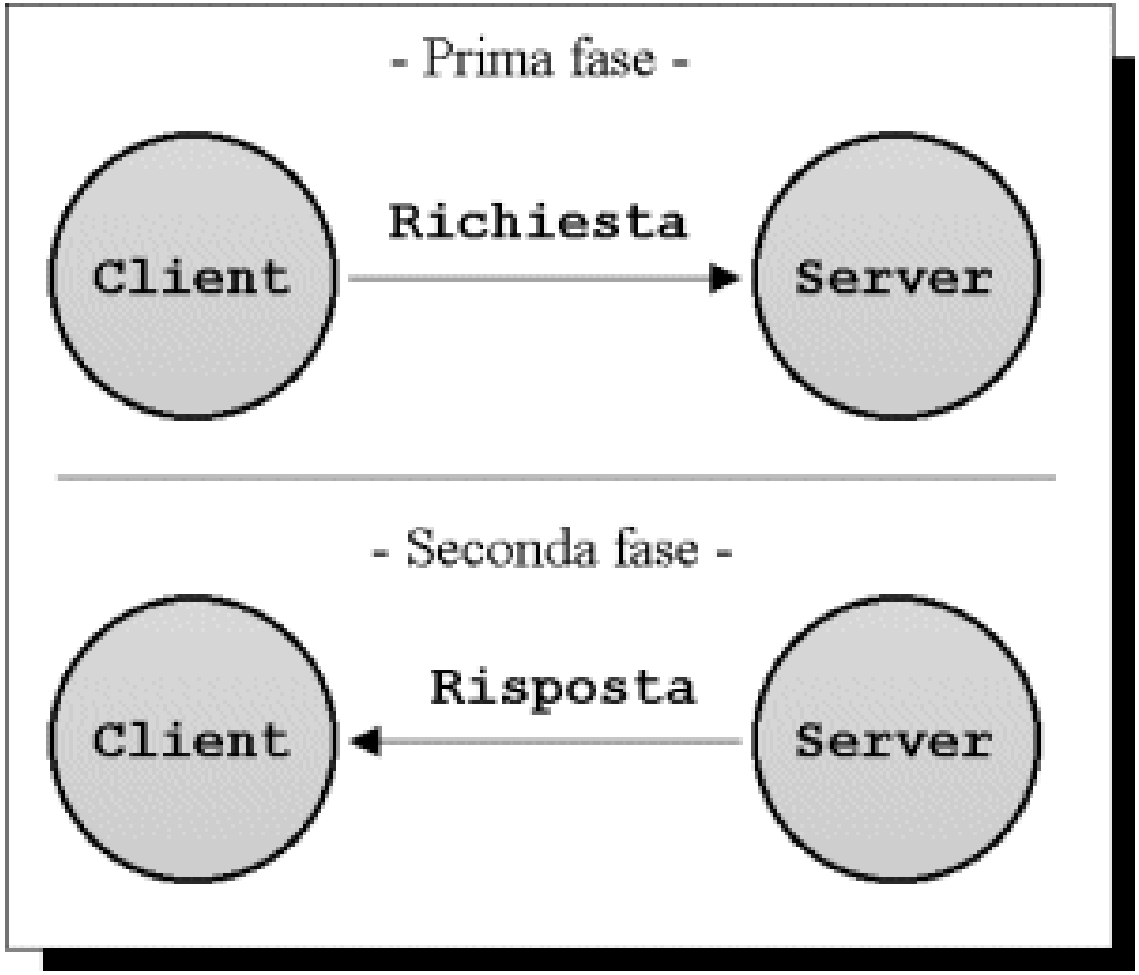
In informatica il termine server indica genericamente un componente o sottosistema informatico di elaborazione che fornisce, a livello logico e a livello fisico, un qualunque tipo di servizio ad altre componenti (tipicamente chiamate client, cioè "cliente") che ne fanno richiesta attraverso una rete di computer, all'interno di un sistema informatico o direttamente in locale su un computer.

Cos’è un Server (2 di 2)

Architettura Client-Server (1 di 4)

In informatica il termine sistema client-server indica un'architettura di rete nella quale genericamente un computer client si connette ad un server per la fruizione di un certo servizio, quale ad esempio la condivisione di una certa risorsa hardware/software con altri client, appoggiandosi alla sottostante architettura protocollare.

Architettura Client-Server (2 di 4)

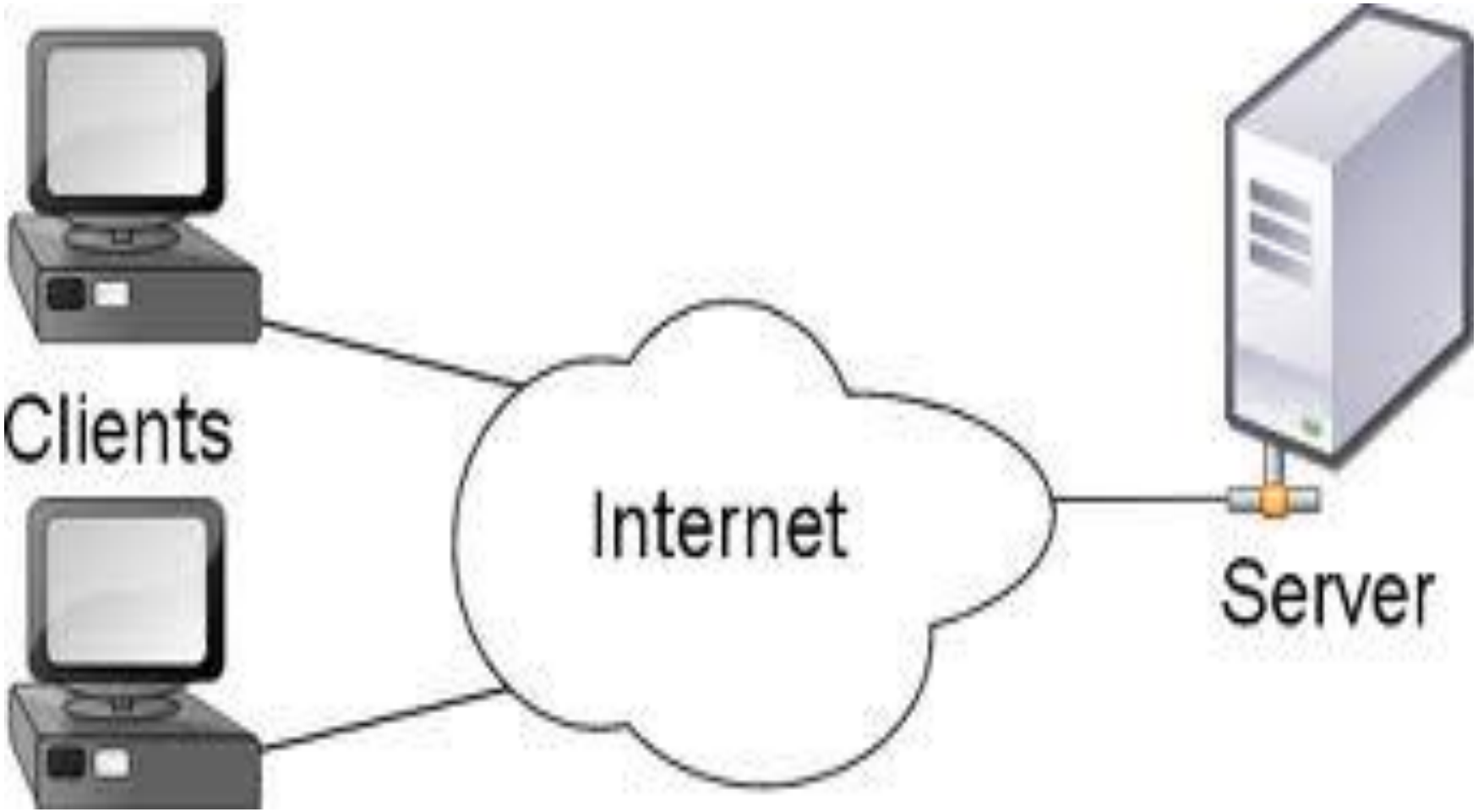


Architettura Client-Server (3 di 4)

Sono organizzati sotto forma di una tipica architettura client-server per la fruizione dei rispettivi servizi:

- Le reti locali aziendali (LAN);
- la rete Internet;
- i sistemi informatici;
- i sistemi operativi.

Architettura Client-Server (4 di 4)



Cos’è un Database (1 di 5)

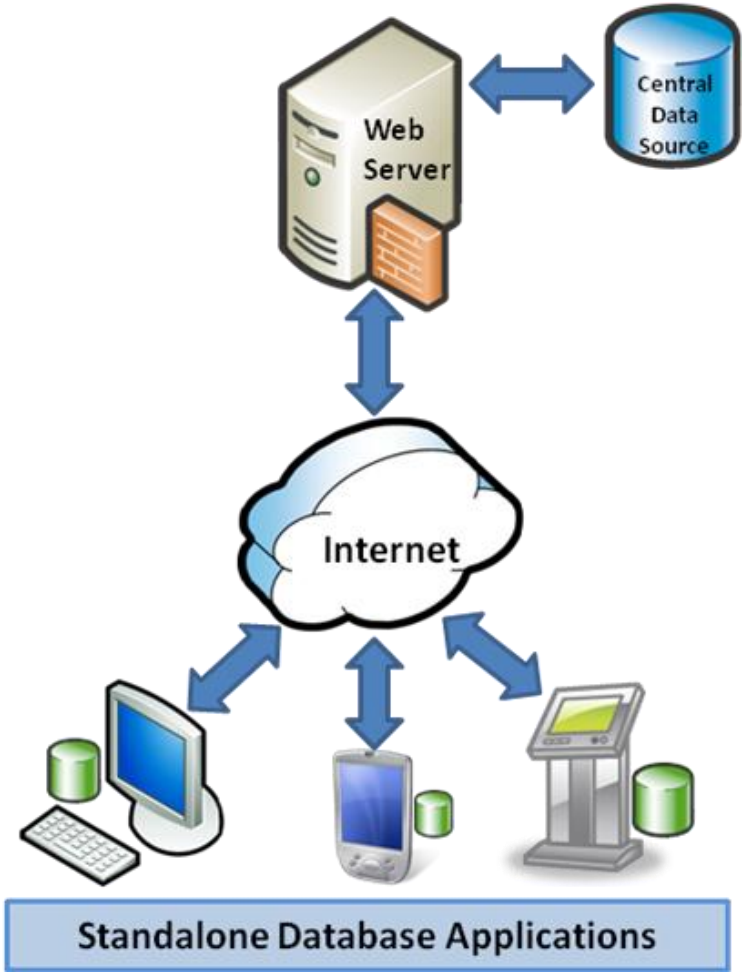
In informatica, il termine database, base di dati o banca dati (a volte abbreviato con la sigla DB), indica un archivio dati, o un insieme di archivi ben strutturati, in cui le informazioni in esso contenute sono strutturate e collegate tra loro secondo un particolare modello logico (relazionale, gerarchico, reticolare o a oggetti).

Cos’è un Database (2 di 5)

Cos’è un Database (3 di 5)

Sono collegate in modo tale da consentire la gestione/organizzazione efficiente dei dati stessi e l'interfacciamento con le richieste dell'utente attraverso i cosiddetti query language (query di ricerca o interrogazione, inserimento, cancellazione, aggiornamento ecc.) grazie a particolari applicazioni software dedicate (DBMS), basate su un'architettura di tipo client-server.

Cos’è un Database (4 di 5)



Cos’è un Database (5 di 5)

Tra i più diffusi DBMS open source troviamo:

- MySQL
- PostgreSQL

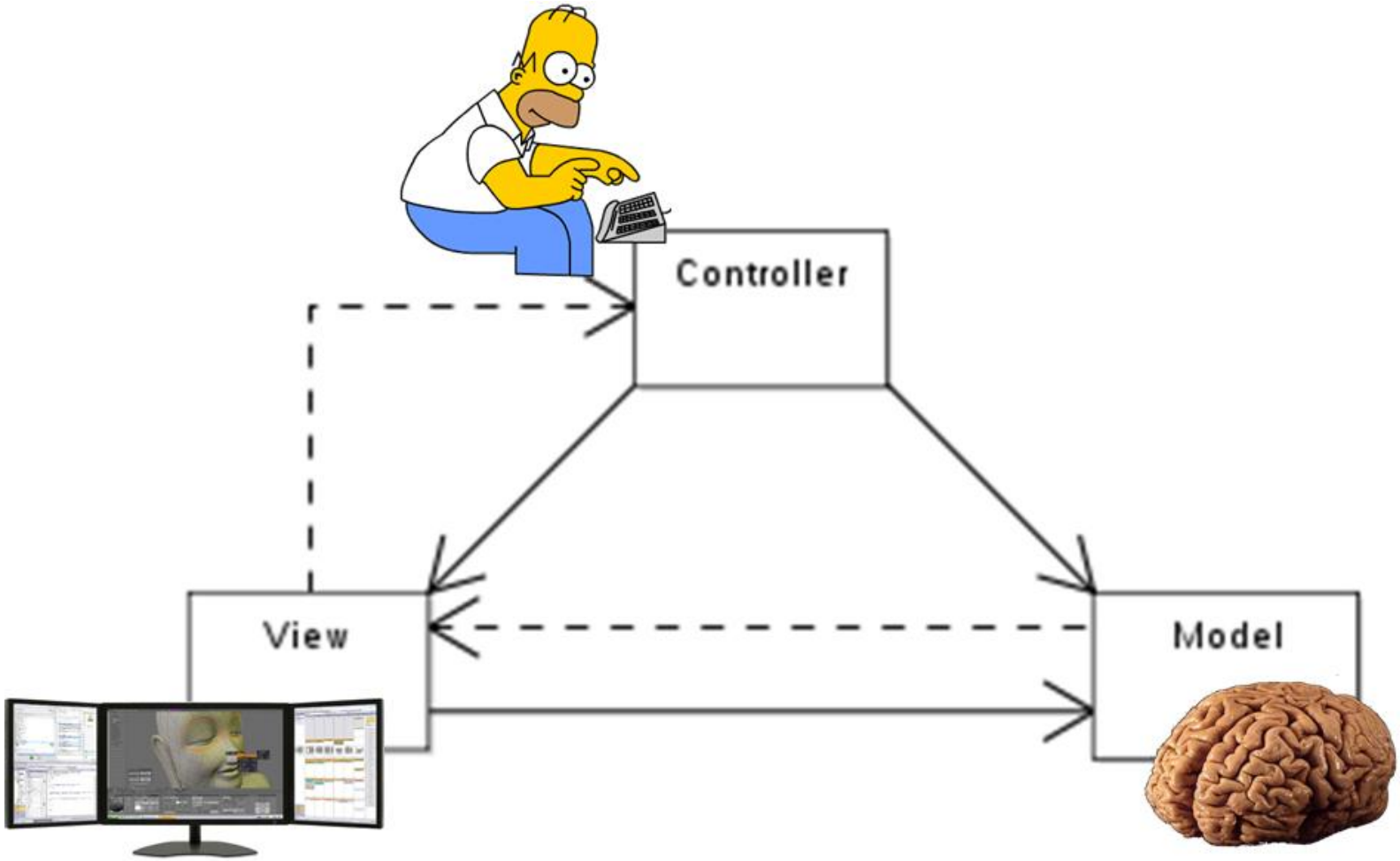
I più diffusi sistemi commerciali sono:

- Oracle
- IBM DB2
- Microsoft SQL Server
- Microsoft Access

Model-View-Controller (1 di 3)

Il Model-View-Controller, in informatica, è un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti, in grado di separare la logica di presentazione dei dati dalla logica di business.

Model-View-Controller (2 di 3)



Model-View-Controller (3 di 3)

Il funzionamento:

- Il browser (tramite il client) invia le richieste;
- Il controller interagisce con il model;
- Il controller chiama la view;
- La view produce la schermata sul browser Funzionamento.

Linguaggi lato Client (1 di 2)

Sono caratterizzate dal fatto che l'esecuzione e l'interpretazione delle istruzioni avvengono in locale sul computer che effettua la richiesta al server;

La conseguenza principale è che una pagina può essere visualizzata solo se gli utenti hanno a disposizione un software in grado di interpretare le istruzioni.

Linguaggi lato Client (2 di 2)

I principali linguaggi e tecnologie che possono essere utilizzati dal lato client sono:

- JavaScript
- Java
- VbScript
- Active X
- Flash
- Jquery

Linguaggi lato Server (1 di 2)

Le tecnologie dal lato server sono caratterizzate dal fatto che permettono l'esecuzione di script sul server, in modo quindi indipendente dall'ambiente di esecuzione dell'utente, tanto da impostazioni e preferenze, quanto dal browser, quanto dal sistema operativo.

Linguaggi lato Server (2 di 2)

Le principali e più comuni tecnologie dal lato server sono:

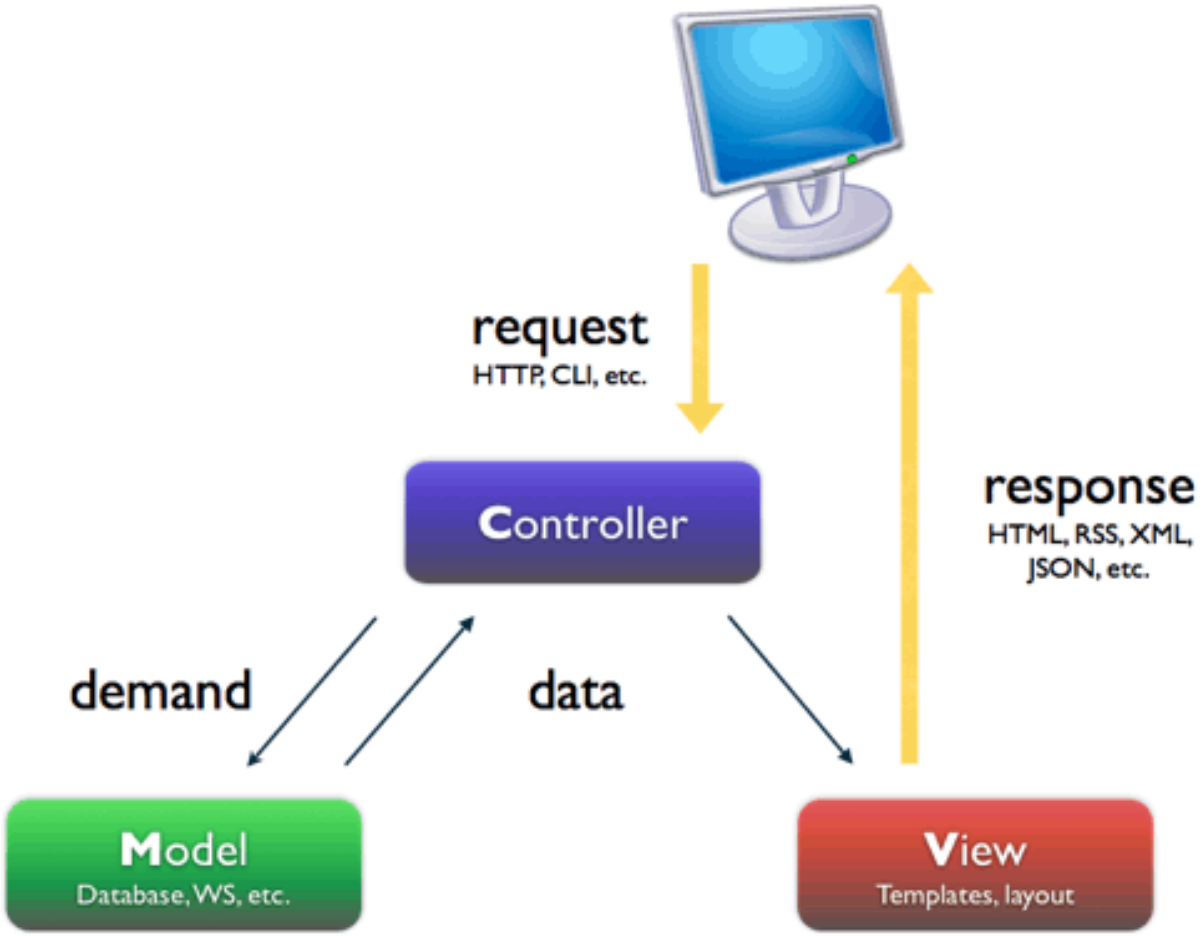
- Common Gateway Interface (CGI)
- Application Program Interface (API)
- Microsoft Active Server Pages (ASP)
- Java Server Pages (JSP)
- Cold Fusion Markup Language (CFML)
- PHP

Cos’è una Pagina Dinamica (1 di 3)

In una Pagina Dinamica i componenti vengono elaborati e composti solo nel momento in cui arriva una richiesta esplicita:

Viene utilizzata nei casi in cui sia necessario generare dei contenuti in modo automatico o in risposta ad un'operazione interattiva effettuata dall'utente;

Cos’è una Pagina Dinamica (2 di 3)



Cos’è una Pagina Dinamica (3 di 3)

Le tecnologie disponibili possono essere e divise in due gruppi principali:

- Linguaggi lato client;
- Linguaggi lato server.

La scelta dell'una o dell'altra avverrà in base a:

- Carico di lavoro da affidare a client e server;
- affidabilità del linguaggio.

Differenza HTML - HTML5 (1 di 3)

HTML 4 e HTML 5 sono entrambi due linguaggi markup;

Il 5 però non è, come immaginabile, il successore del 4 ma e destinato a diventare il nuovo standard;

Infatti, prima dell'HTML 5, lo standard proposto era l'XHTML (Extension HTML).

Differenza HTML - HTML5 (2 di 3)

HTML 4 è stato reso disponibile a natale del 1999 e prevedeva una prima reale adozione dei fogli di stile (CSS) per separare la struttura dallo stile.

Dopo il fallimento della seconda release di XHTML è iniziata la concezione di HTML 5.

Differenza HTML - HTML5 (3 di 3)

Quest'ultimo linguaggio si propone di migliorare la possibilità di strutturazione delle informazioni sul Web, oltre che accentuando la separazione tra stile, contenuti e struttura, anche tramite la creazione di nuovi tag come <video> e <audio> che meglio si adattano alla nuova evoluzione del WEB.

Cos’è un Framework (1 di 4)

In informatica, e specificatamente nello sviluppo software, un framework è una struttura logica di supporto (spesso un'implementazione logica di un particolare design pattern) su cui un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore.

Cos’è un Framework (2 di 4)

Alla base di un framework c'è sempre una serie di librerie di codice utilizzabili con uno o più linguaggi di programmazione, spesso corredate da una serie di strumenti di supporto allo sviluppo del software, come ad esempio un IDE, un debugger o altri strumenti ideati per aumentare la velocità di sviluppo del prodotto finito.

Cos’è un Framework (3 di 4)

L'utilizzo di un framework impone dunque al programmatore una precisa metodologia di sviluppo del software;

Quando si parla di Framework è doveroso ricordare il pattern MVC.

Cos’è un Framework (4 di 4)

I framework MVC, più diffusi, sono:

- Zend Framework (PHP)
- Ruby on Rails (Ruby)
- Symfony (PHP)
- Flex (Java)
- ASP.NET (ASP e C#)
- Yii (PHP)
- CakePHP (PHP)

Cos’è un Framework CSS (1 di 2)

Un framework è una struttura logica di supporto su cui un software può essere progettato e realizzato spesso facilitandone lo sviluppo;

Nel mondo del web design, per essere più specifici, un framework è l’insieme strutturato di file (HTML, CSS e Javascript), cartelle e codice standardizzato secondo una logica, che viene utilizzato come fondamenta o supporto per la creazione di un nuovo sito o applicazione web.

Cos’è un Framework CSS (2 di 2)

I framework CSS, più diffusi, sono:

- Bootstrap;
- Blueprint;
- 960 grid;
- YUI CSS;
- angularJS .

Bentobox (1 di 6)

Che cosa abbiamo imparato fino ad ora?

- La differenza tra client e server;
- La differenza tra un linguaggio lato cliente e un linguaggio lato server;
- La differenza tra un sito web e una applicazione web;
- La differenza tra un linguaggio di programmazione e un web framework;
- Regole di copia-incolla (Google everything ;-))

Bentobox (2 di 6)

Bentobox (3 di 6)

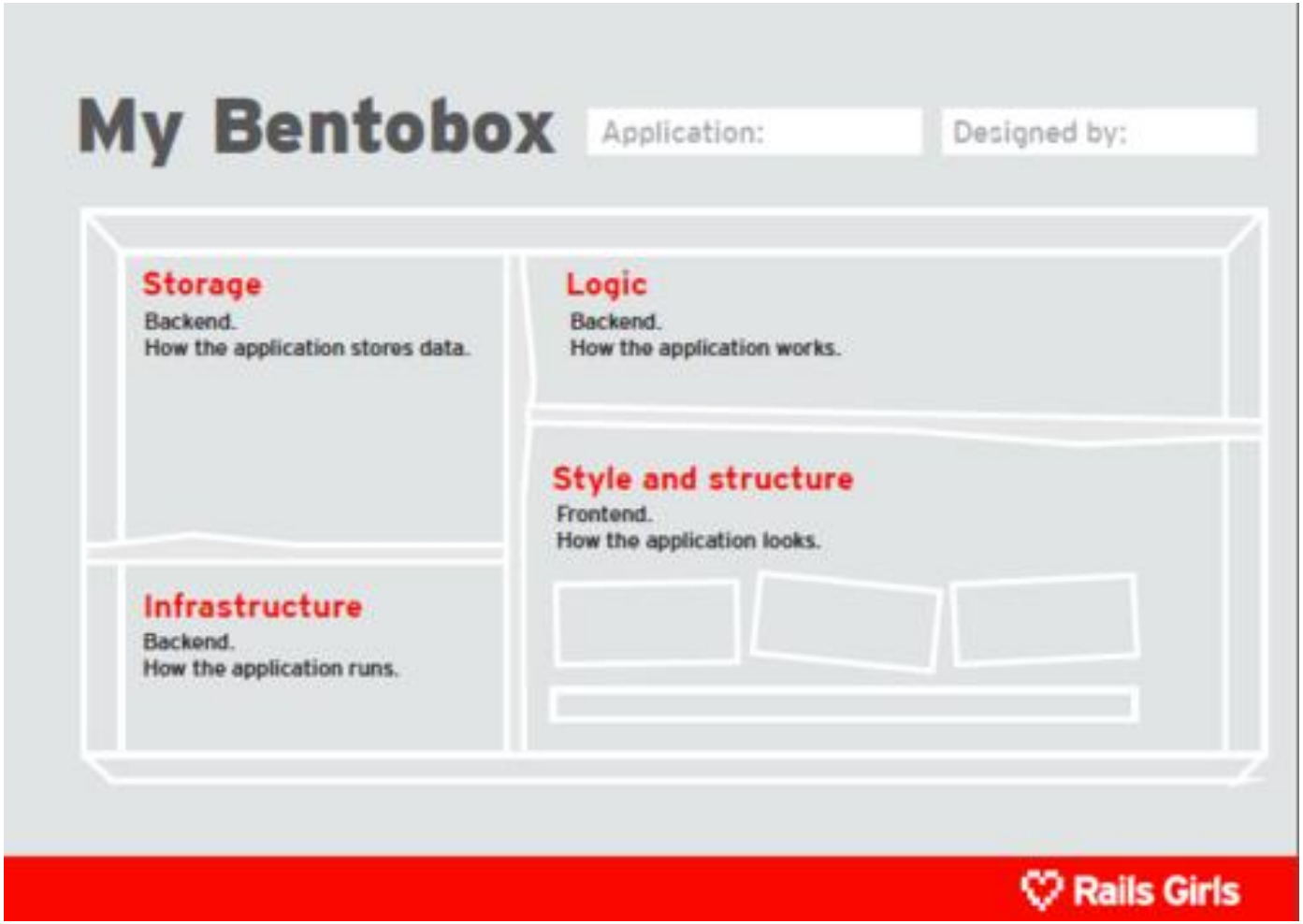
Perché Bento?

Il **bentō** (お弁当 *obentō*) è una sorta di vassoio contenitore con coperchio di varie forme e materiali contenente un pasto, in singola porzione, impacchettato in casa o comprato fuori, comune nella cucina giapponese;

il cui contenuto è disposto nel modo più efficiente possibile e in maniera graziosa;

È a tutti gli effetti un puzzle ;-)

Bentobox (4 di 6)

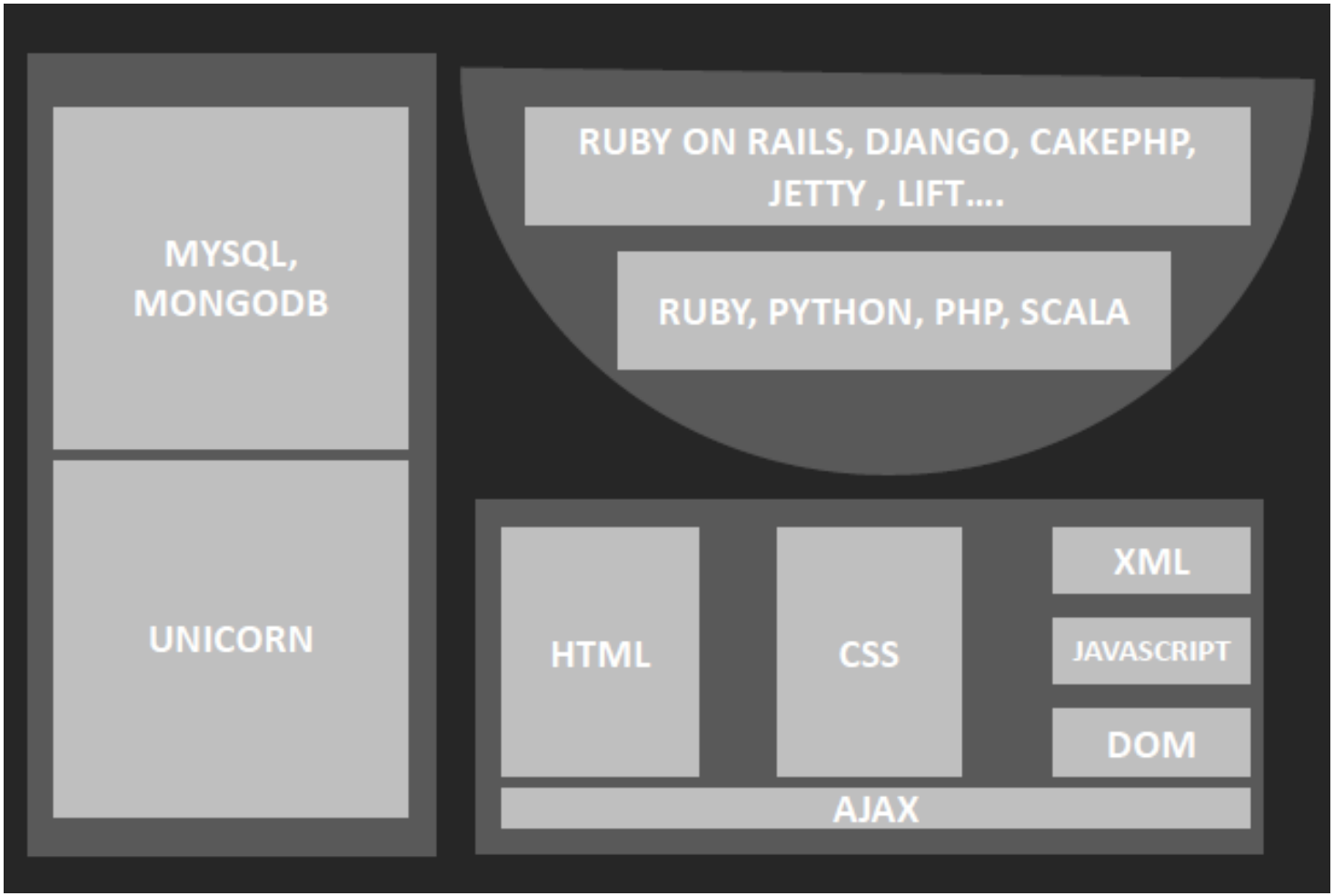


Bentobox (5 di 6)

Ora tocca a voi comporre la vostra Bento Box:

- Storage
- Infra
- The Logic
- The Structure e Style

Bentobox (6 di 6)



Cos’è Ruby on Rails (1 di 2)

Ruby on Rails (spesso chiamato RoR o semplicemente Rails) è un framework open source per applicazioni web scritto in Ruby da David Heinemeier Hansson per conto della 37signals;

la sua architettura è fortemente ispirata al paradigma Model-View-Controller (MVC).

Cos’è Ruby on Rails (2 di 2)

Ruby on Rails è :

stato creata nel 2005 e da allora ha acquisito una trazione globale impressionante;

una piattaforma open-source di sviluppo web, realizzata con il linguaggio di programmazione Ruby;

stato sviluppato implementando con particolare attenzione le funzionalità destinate all'interazione del framework con le basi di dati e le informazioni in esse archiviate.

A cosa serve Ruby on Rails (1 di 2)

Serve a sviluppare applicazioni in modo semplice diminuendo la percentuale di codice che solitamente va a ripetersi nelle applicazioni.

N.B.: Una grande contributo è dato dal pattern Model View Controller che è una pratica di programmazione che semplifica la separazione tra presentazione dei dati, logica della app e contenuti.

A cosa serve Ruby on Rails (2 di 2)

Il risultato è che le app sviluppate in Ruby on Rails sono particolarmente indicate per progetti dinamici, flessibili che necessitano aggiornamenti continui o ampliamenti futuri.

Framework CSS utilizzati con Rails

È possibile, per RoR, fare riferimento a :

- Bootstrap;
- Blueprint;
- 960 grid;
- YUI CSS.

Database utilizzati con Rails (1 di 2)

Tra i numerosi DBMS supportati da RoR, per citare soltanto alcuni di quelli rilasciati sotto licenza Open Source, è possibile fare riferimento a :

- MySQL;
- SQLite;
- PostgreSQL;
- MongoDB.

Database utilizzati con Rails (2 di 2)

database.yml

```
1  # SQLite version 3.x
2  #   gem install sqlite3-ruby (not necessary on OS X Leopard)
3  development:
4    adapter: sqlite3
5    database: db/development.sqlite3
6    timeout: 5000
7
8  # Warning: The database defined as 'test' will be erased and
9  # re-generated from your development database when you run 'rake'.
10 # Do not set this db to the same as development or production.
11 test:
12   adapter: sqlite3
13   database: db/test.sqlite3
14   timeout: 5000
15
16 production:
17   adapter: sqlite3
18   database: db/production.sqlite3
19   timeout: 5000
20
```

Model-View-Controller (1 di 5)

Le applicazioni sviluppate con Rails hanno una peculiarità, ovvero sono tutte organizzate secondo una struttura comune;

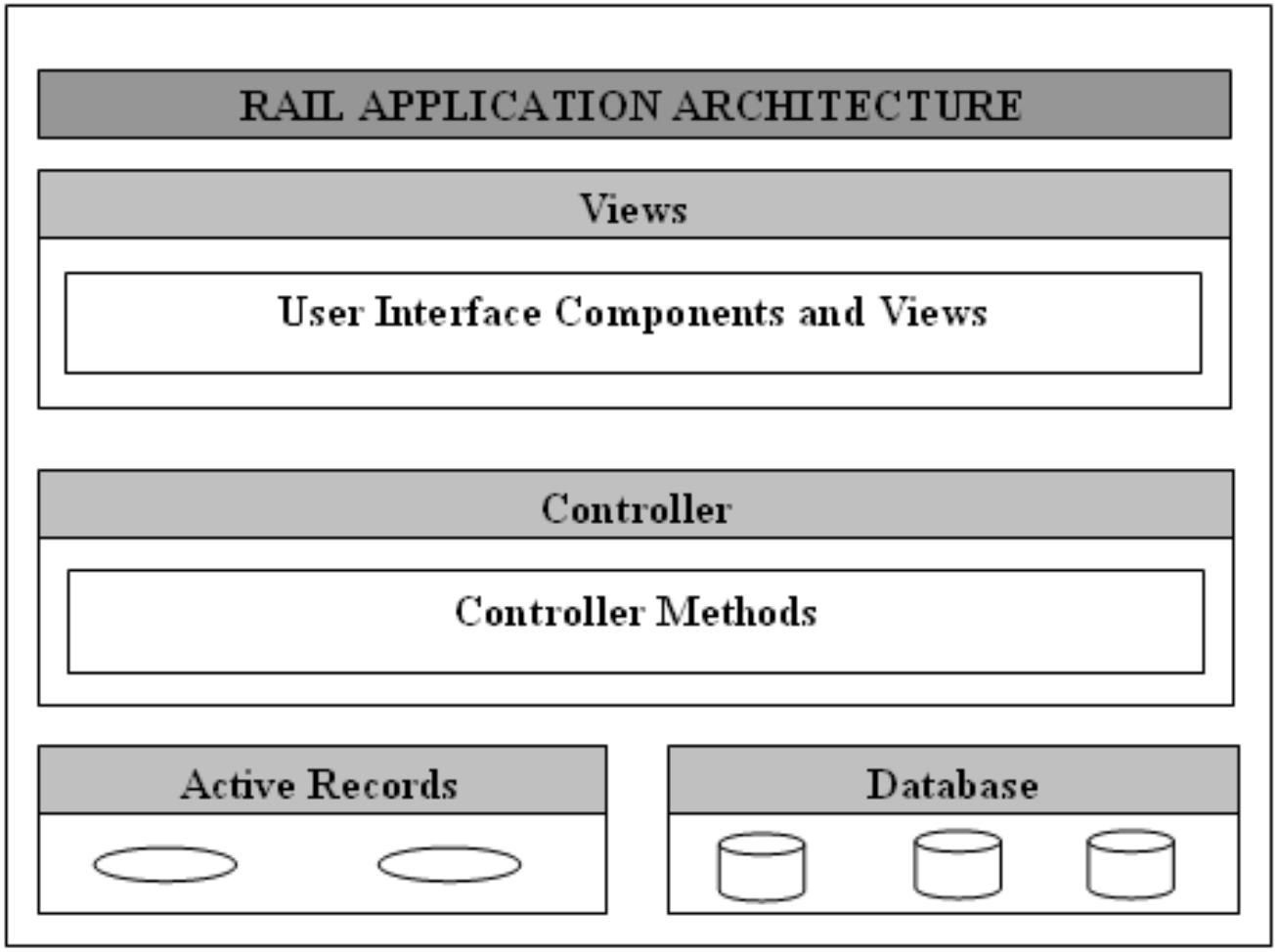
Questa è una conseguenza del fatto che il comando rails genera una serie di directory e file che forniscono una certa linea guida nello sviluppo, linea che se rispettata permette a Rails di effettuare molte cose automaticamente.

Model-View-Controller (2 di 5)

Esempio : caricare i file, generarli ed individuarli a runtime e molto altro;

Questa struttura comune permette anche di comprendere con semplicità il codice di progetti realizzati da altri, in quanto sono organizzati nella stessa maniera.

Model-View-Controller (3 di 5)

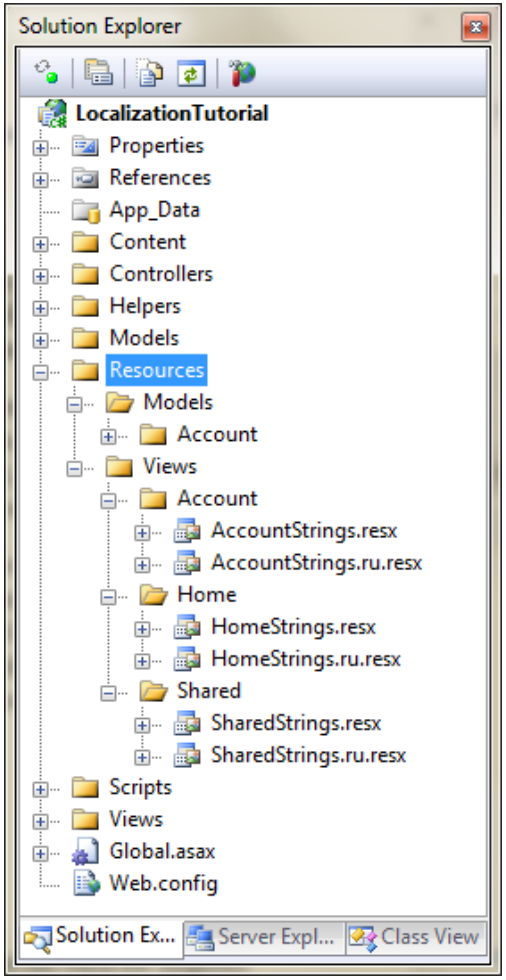


Model-View-Controller (4 di 5)

I dati (model) sono separati dall'interfaccia utente (view):

- Model
  - Accesso ai dati e alla logica di business
  - Indipendente dalla View e dal Controller
- Controller
  - Coordina l'interazione tra l'utente, le View, e il Model
- View
  - Presentazione dei dati e interazione con l'utente
  - Accesso in sola lettura al Model

Model-View-Controller (5 di 5)

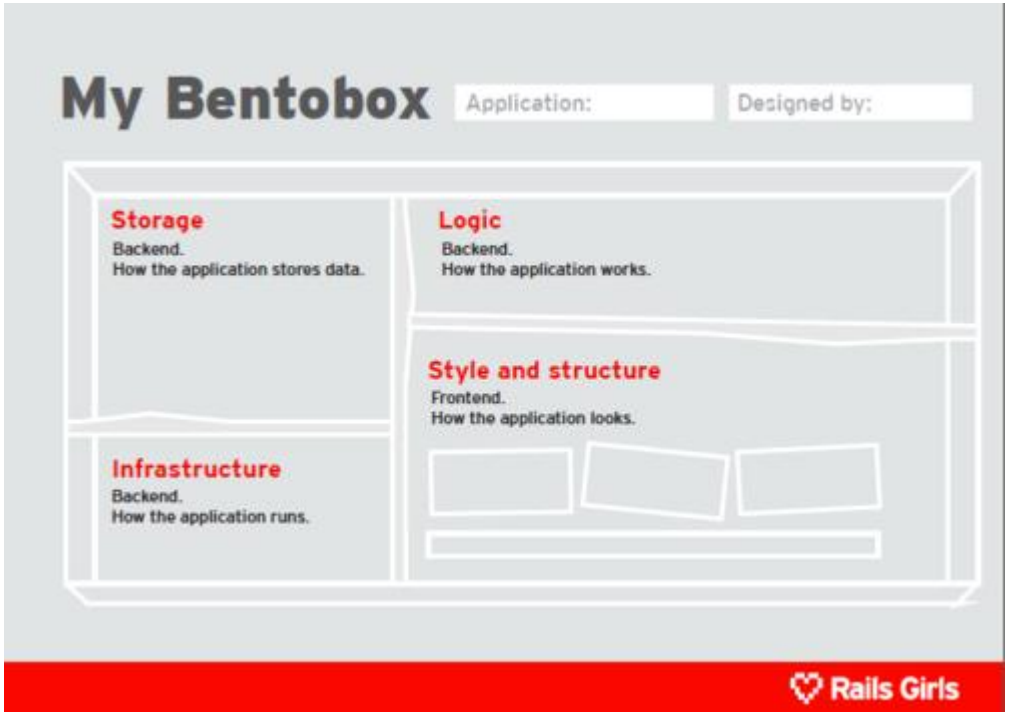


In conclusione (1 di 1)

Vediamo ora il risultato della nostra lezione:

```
rails new rails_girls
```

Fine



Grazie per l’attenzione