

Open Web Technology

Presented at the Seminar Organised at NIC-Hq, New Delhi on 20/03/2014.

Open Technology Group,
National Informatics Centre,
DeitY, MCIT, Govt. of India,
NIC-Taramani, STPI Campus,
First Floor, No. 5, Rajiv Gandhi Salai,
Taramani, Chennai-600113.
e-mail:- otc@nic.in

NIC

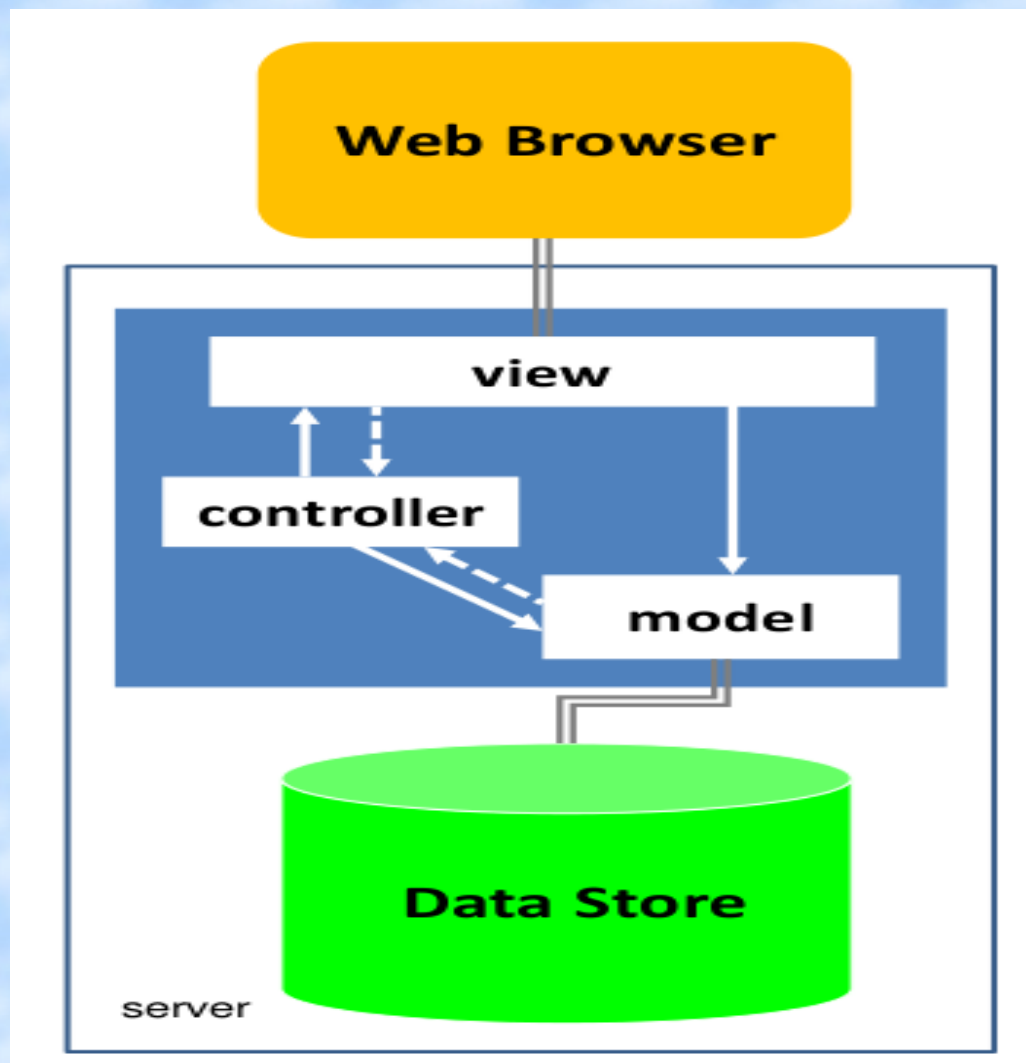
NATIONAL
INFORMATICS
CENTRE



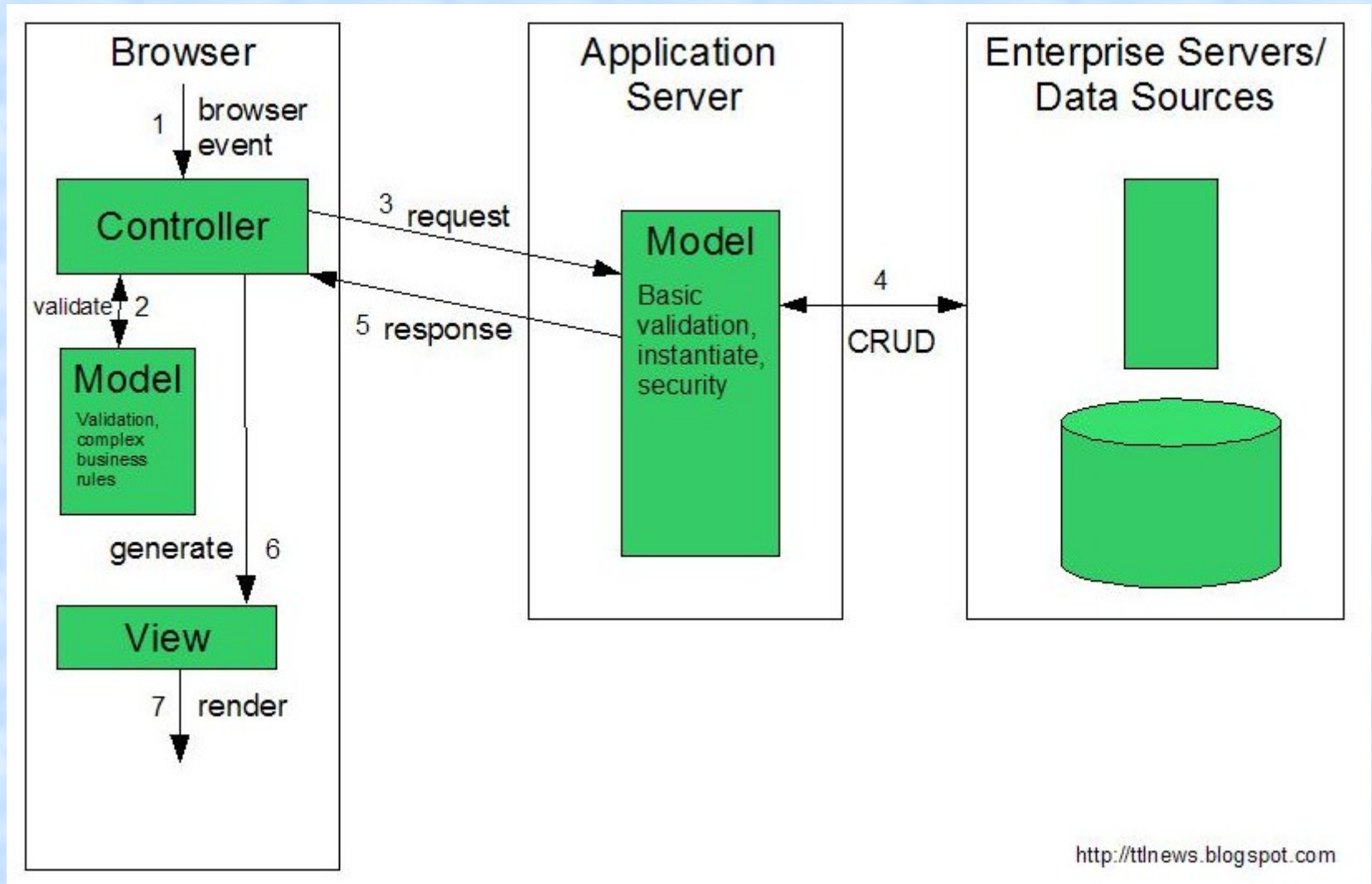
Existing Challenges

- **UnReliable Network:** Need to work in Unreliable Network Connectivity
- **Multiple-Delivery Chennals:** Need to port/re-develop the application on variety of devices (mobiles, tablets, ipTV, Point-of-Sales Devices, etc)
- **Over-use of Servers:** Waste of Computational Resources (CPU & Memory) at Server-Side makes the procurement of more numbers of powerful-servers.
- **Poor-use of Clients:** Under-utilisation of Computational Resources at Client-Side (like desktops, mobiles, tablets)
- **Complex Data Models:** Use of Complex Data Models (Name-Value Pair, Object, Relational, XML) and their conversions across multiple (web-browser, app-server, db-server, web-service) layers
- **High Skill sets:** Need for High Level of Skill Sets to learn, develop and manage the e-Governance applications using non-standard approaches / technologies, operating systems, separate language for each specific-device.

Normal / Conventional Server-Centric Web Applications

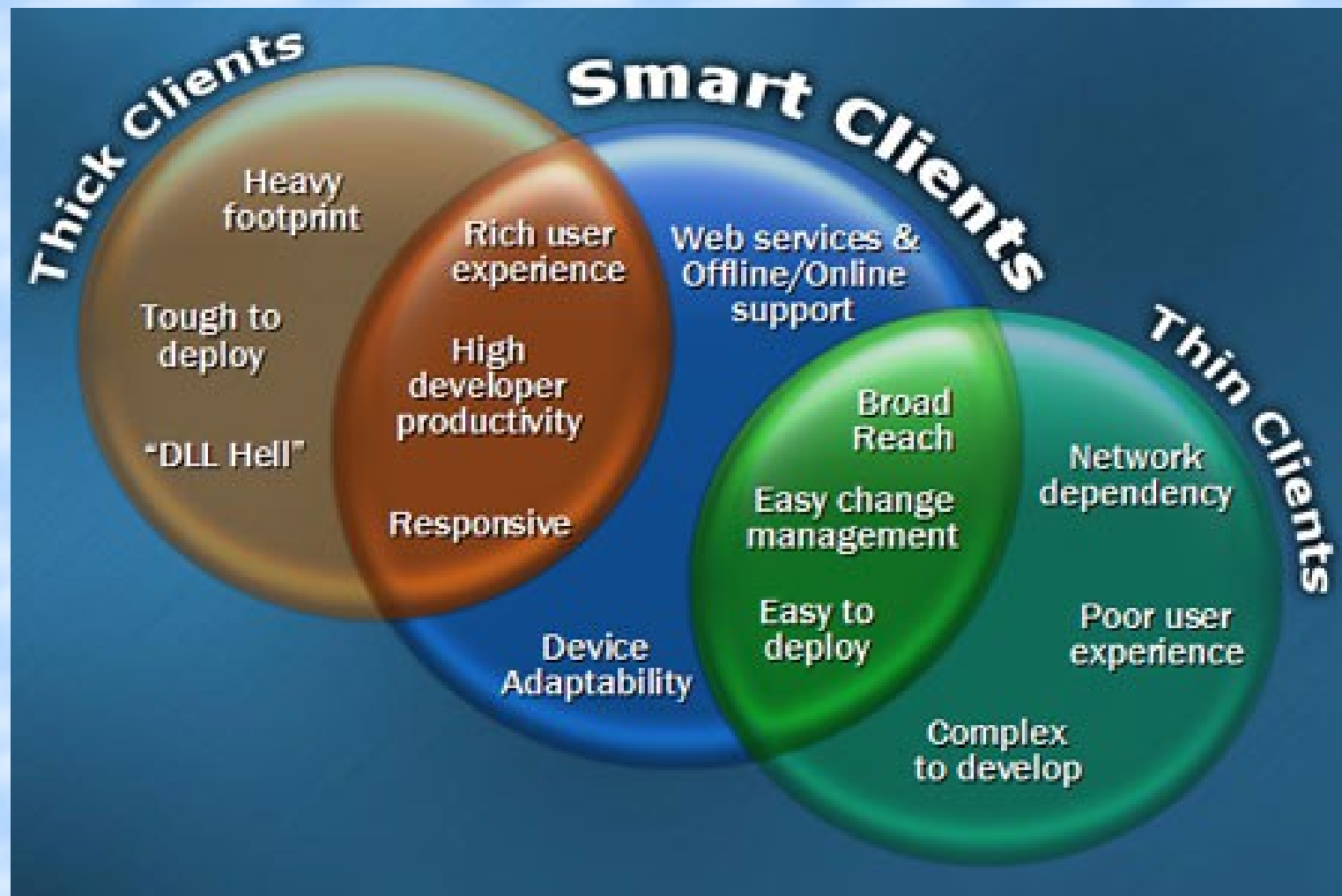


New Client-Centric Applications



Rich Web Applications - Java-Applet, Flash, ActiveX
Rich Applications – Windows-Smart-Client, Java-Web-Start

Rich / Smart / Thick Client Web Application



Mobile Revolution and its Impact

- The explosion of varieties and types of mobiles, especially smart-phones with HTML5 browser, challenged native mobile applications adoption.
- In 2011, there were about 336 million HTML5 capable mobiles sold.
- As per the report, Research firm Strategy Analytics forecasts that one billion HTML5-capable mobile devices will be sold in 2013.
- ABI Research sees more than 2.1 billion mobile devices with HTML5 browsers by 2016.
- IDC estimates indicate that over 80 percent of all mobile applications will be wholly or partly based on HTML5 by 2015.
- The application development has been drastically changed due to the emergence of HTML5 based Open Web Technology (OWT).

Internet of Thing (IoT) / Web of Everything Revolution (1)

- One or two mobile devices are considered per person; i.e. billions of devices for the billions of people. Most of conventional computations are obsolete now !
- Whereas, Internet of Thing (IoT) / Web of Everything (WoE) is a paradigm shift in technology and lifestyle, a shift from a disconnected World to an always-connected and always-on World by using Intelligent-Sensors. It's a world where every device from lifestyle apparel, utility meters, cars, appliances, to windows and doors are connected. It is invading even fields like agriculture, health.

Internet of Thing (IoT) / Web of Everything Revolution (2)

- **Several devices per person are anticipated which will result in trillions of devices for billions of people. (Germany declared that IoT / WoE is the Fourth Industry Revolution.).**
- **IoT / WoE is emerging based on Open Web Technology, NoSQL database, Near-Real-Time Analytics using NoSQL, etc.**

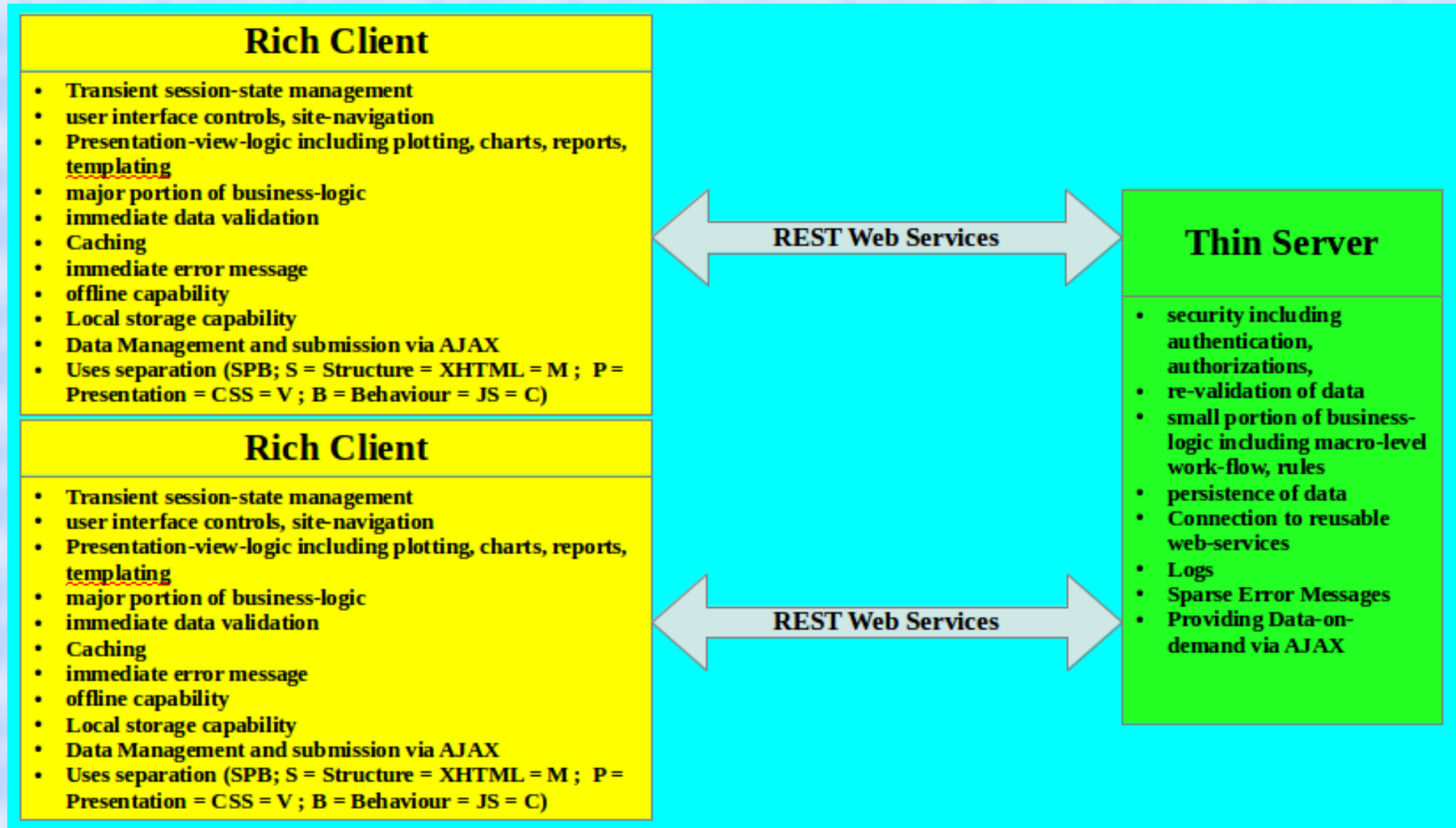
Emergence of Browser as a Platform (1)

- Most types of *mobiles* (feature-phones / smart-phones) and *tablets support web browsers*; web browser is also available on recent *hand-held devices* (like *Point-of-Sales*) and *ipTV*.
- The web browser *offers conventional features* (like caching, offline, local storage, files, graphics, audio, video, network communications, etc) which were offered earlier by traditional desktop.
- The web browsers start *adhering to standards* / components (like HTML5, CSS3, JavaScript, REST Web Service, JSON Data). The server interacts with the browser-client mostly through REST web services or new methods.

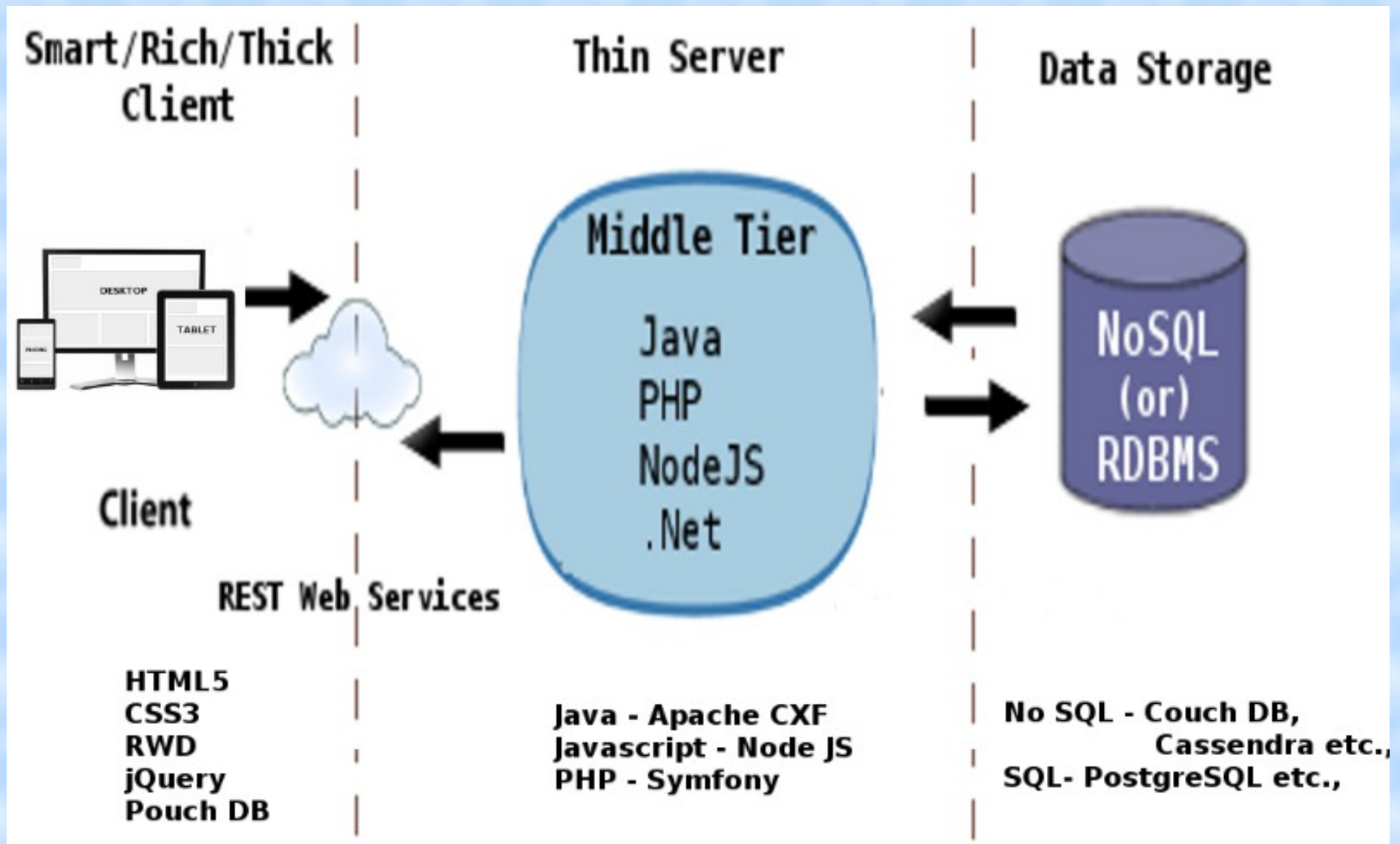
Emergence of Browser as a Platform (2)

- *Most of generic-features needed for e-Gov applications are available on recent browsers* and hence *same application runs on browsers available on multiple-devices* (if any mobile specific features are required, then libraries like PhoneGap / Apache-Cordova can be used to make the web application to run on all common mobile devices).
- Application can also *work in unreliable networks* since *browser has off-line capabilities and local-caching*; browser can sync the information from the local storage to remote server whenever network is available.

Common Functions of Rich-Web-Client and Thin-Server



Basics of Open Web Technology (1)



Basics of Open Web Technology (2)

Open Web Technology (OWT) is an *umbrella term*, refers to a *collection of technologies* like HTML5, CSS3, JavaScript, RESTful Web-Services, JSON Data, etc.

The OWT facilitates *Web Browsers as Platform* for innovation, consolidation and cost efficiencies.

The OWT offers the transformation of the *server-centric* applications to a *client-centric one*.

This would obviously help in *reducing the overhead on server and network-traffic*.

Basics of Open Web Technology (3)

HTML5 Connectivity Revolution

- **WebSocket**

- It sits along side HTTP and offers bi-directional, full-duplex asynchronous communications channels, over a single Transmission Control Protocol (TCP) socket. It is designed for low latency and messages with very little overhead.

- **Server-Sent Event**

- It allows a server to push events to client.

- **WebRTC**

- WebRTC (Web Real-Time-Communication) supports browser-to-browser applications for voice calling, video chat, and P2P file sharing without plugins.

Basics of Open Web Technology (4)

HTML5 Connectivity Revolution

- **REST Web Service / Ajax / XMLHttpRequest**

- By using the AJAX approach, Web Server can be contacted independently of the View - behind the scenes of the current "View". New possibilities are, (i) Loading of Partial Views (ii) Validating User-inputs continuously or on-demand, without having to submit the entire "Form" (iii) Loading of New menu options or List-box values dynamically. All of these techniques save considerable amounts of time, and enhance the speed of the application.

- **WebWorker**

- It is a JavaScript script executed from an HTML page that runs in the background, independently of other user-interface scripts that may also have been executed from the same HTML page. Web workers are able to utilize multi-core CPUs more effectively.

How to overcome the Challenges (1)

Develop Once and Run Everywhere Approach: The mobile revolution forced most of the companies to offer their applications to run on popular web browsers;

- ***works on variety of devices (like mobile, tablet, desktop, ipTV) and multiple operating systems using the same source-code-base***
- ***works in unreliable networks by using off-line capabilities***
- ***consumes maximal resources in the client***
- ***uses browser as a platform without plug-ins and offers conventional features like desktop; in case, browser does not support specific-features like access to the users client-device-specific features, then the “Hybrid-Approach” using Apache-Cordova / PhoneGap libraries can be preferred while retaining unified development.***

How to over come the Challenges (2)

Develop Once and Run Everywhere Approach:

- *uses lesser server-resources and hence fewer servers are required.*
- *adheres to Open Web Technology Standards like HTML5, CSS3, ECMA-Script/JavaScript, JSON, WebSocket, WebWorker, ServerSentEvent, WebStorage, AMD/CommonJS/Harmony, etc. Hence, the application developed has the potential to last long.*
- *offers simpler software development, easy maintenance and quicker response due to the option to use same JSON data-model across web-layers like browser, application-server, web-service and database.*
- *requires lower skill-sets compared to the skill-sets needed for the existing conventional (server-centric) and device-specific approaches.*
- *Client side code can immediately react to user input, rather than waiting for network transfers.*

Approach to Adopt (1)

- ***Skill Development programs (Technical Courses, Awareness Programs, and Workshops).***
- ***Preparation and sharing of guidelines / documents / tutorials***
- ***Creation, maintenance and sharing of repository of sample templates, forms, reports, dashboards, components, modules, etc.***
- ***Provisioning of readily-built development & deployment environments with DR facilities using virtualisation / cloud solutions.***

Approach to Adopt (2)

- *Establish/ Enhance/sustain the ecosystem (including industry, academia, community) for support services on the stacks.*
- *Form Working Groups to develop course-ware and introduction in select Institutions.*
- *Identify and implement visible E-Gov projects at State/ Centre levels.*
- *A set of guidelines on the inclusion in the procurement / tender.*

Open Source Software Stack (1)

Functional Areas for Tools (Minimal / Primary)	Open Web Technology Stack (Minimal / Primary)
Programming Languages on Client-side for Building Generic-Mobile and Desktop Solutions	<i>HTML5, CSS3, JavaScript, JQuery</i>
Relational Database	<i>PostgreSQL</i>
Web Service Framework	<i>Apache CXF with Apache Tomcat & Apache HTTP Server</i>
Programming Language on Server-side and Library	<i>Core Java with OpenJDK</i>

Open Source Software Stack (2)

Functional Areas for Tools (Additional / Optional)	Open Web Technology Stack (Additional / Optional)
Building Mobile-Native (OS-Android, iOS, Windows Phone, BlackBerry, Symbian)	<i>Apache-Cordova</i>
Portal/CMS	<i>Drupal</i>
Non-Relational Database	<i>Apache CouchDB</i>
Virtualisation	<i>Xen Server</i>
Cloud Platform	<i>Apache-CloudStack / OpenStack</i>

Use Case scenarios (1)

where

- *immediate user response is required with better user interface control facilities (since presentation-logic & business-logic are handled at the client; need not wait for network transfers; client-side caching is possible)*
- *high server-scalability is required when potentially very large numbers of customers access the application (since server handles mainly the security and persistence of data; presentation-logic & business-logic are handled at the client)*

Use Case scenarios (2)

where

- *session-state is to be maintained for longer duration due to (i) complexity of business rules, (ii) form-filling, (iii) office productivity applications, (iv) business / data analysis applications, (v) media applications, (vi) graphics applications, (vii) online mapping, (viii) personal information and communication, (ix) long-running transactions, etc. (since session state management is shifted to the client and stateless server with REST services is deployed)*

Use Case scenarios (3)

where

- *offline capability is required (due to clean separation between client and server which makes it simpler to implement offline modes; lighter database like PouchDB / HSQLDB can be considered at client for easy administration)*
- *cloud enabling capability is required (due to clean separation between rich-browser-client and thin-server which makes it simpler to implement cloud enabling capability)*

Recommendation (1)

Open Web Technology should be preferred to develop once and run the same on all devices. Device Specific Development (Desktop, Tablet, Mobile, etc.) should be discouraged since

OWT

- *Requires lesser number of servers since it uses the maximal client resources.*
- *Reduces the learning effort required by developers to support varied devices.*
- *Enables same application to run on varied devices like mobiles, tablets, desktop, ipTV.*

Recommendation (2)

..... OWT

- *Even in unreliable network, the application works since it has offline capabilities.*
- *Uses lesser network bandwidth, mainly for the data synchronisation which leads to reduced network traffic.*
- *Use of simplified data model leads to faster software development as well as quicker operations at all web-layers.*
- *Long lasting application since the approach is based on open standards.*

Thank You !

Sharing of Some Thoughts from

Dr. P. Balasubramanian,

Email - p.balu@nic.in