Experience of enriching Tizen HTML5 application

Jingfu.Ye@intel.com

TIZEN™ DEVELOPER SUMMIT 2014 SHANGHAI

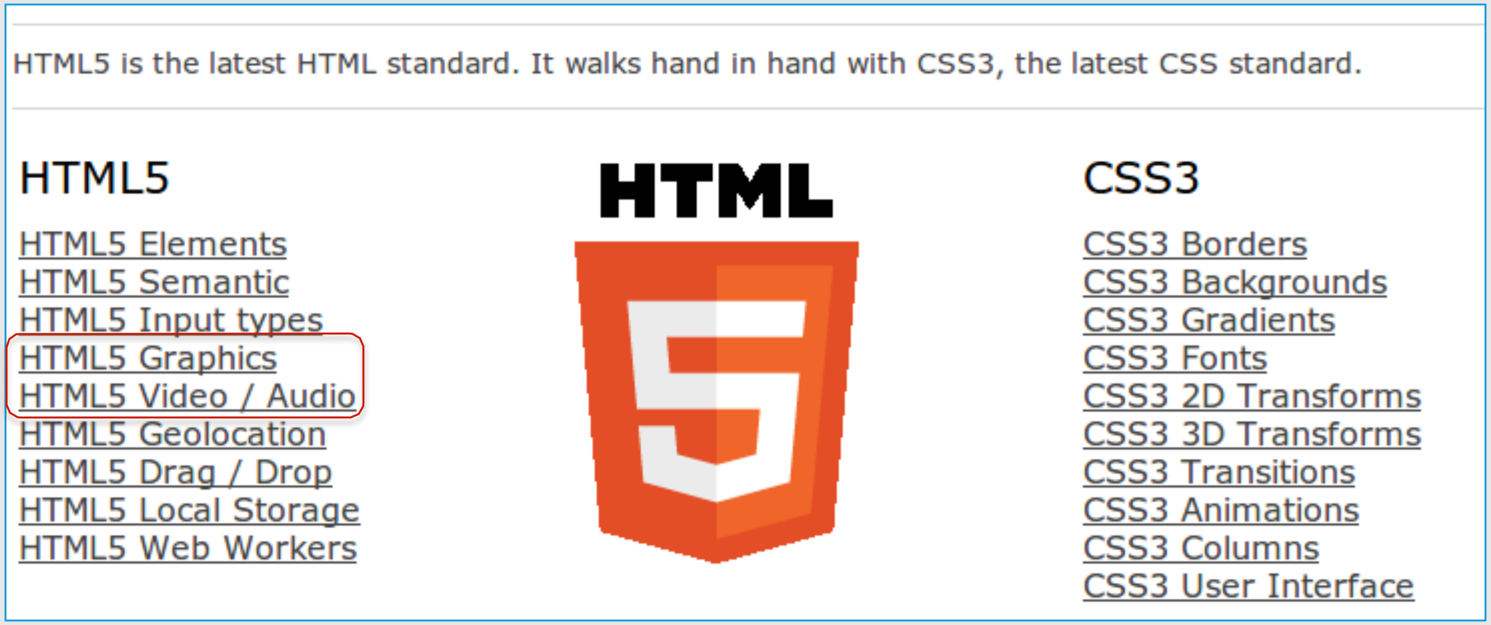
TIZEN开发者峰会（上海）

Outline

- Background introduction
- Graphics capability
- Collaboration
- Real-time communication
- Concurrent Note
- Customization for Tizen
- Summary

Background introduction

HTML5 New Features



HTML5 is the latest HTML standard. It walks hand in hand with CSS3, the latest CSS standard.

HTML5

- HTML5 Elements
- HTML5 Semantic
- HTML5 Input types
- HTML5 Graphics
- HTML5 Video / Audio
- HTML5 Geolocation
- HTML5 Drag / Drop
- HTML5 Local Storage
- HTML5 Web Workers

CSS3

- CSS3 Borders
- CSS3 Backgrounds
- CSS3 Gradients
- CSS3 Fonts
- CSS3 2D Transforms
- CSS3 3D Transforms
- CSS3 Transitions
- CSS3 Animations
- CSS3 Columns
- CSS3 User Interface

www.w3schools.com/html/html5_intro.asp

Benefits of Using HTML5

- **Zero installation & timely upgrade**

  It makes app more accessible and feasible and updates can be performed in background and affect immediately
- **Offline cache**

  Allow app to work even when not connected to internet.
- **Graphics & audio & video**

  Strengthen app with rich cool features in standard way, without plugin.
- **Cross-platform & cross-browser**

  Write once, run everywhere!
- **Hybrid application on mobile**

  Win-win when combine HTML5 with mobile development.
- **Websocket**

  Allow client to have full-duplex communication with server.

Graphics capability

Canvas introduction

HTML5 has the tag <canvas> which contents are rendered with JavaScript. In order to leverage the HTML5 canvas we need to place the <canvas> tag inside of HTML document. Then we can access the canvas in JavaScript and then utilize the canvas API to draw visualizations.

As more browsers offer hardware acceleration for the HTML5 canvas tag, it'll become easier to build very fast complicated animations on the fly with JavaScript. And more development tools will crop up, making it easier to create canvas animations.

Canvas element has canvas **context** which is an object with properties and methods that you can use to render graphics inside the canvas element.

The canvas context can be 2D and 3D(WebGL).

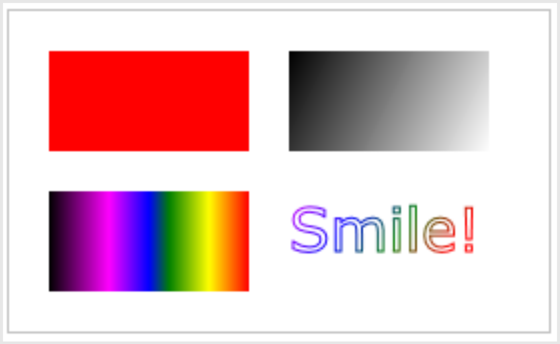
Canvas 2D

HTML5 <canvas> element is used to draw 2D graphics on a web page.
Canvas has methods for drawing paths, boxes, circles, text and images.

```
// HTML5 element
<canvas id="myCanvas" width="200" height="100">
Your browser does not support the HTML5 canvas tag.
</canvas>

// Javascript
<script>
var c = document.getElementById("myCanvas"); var ctx =
c.getContext("2d");
var grd = ctx.createLinearGradient(0,0,200,0);
grd.addColorStop(0,"red"); grd.addColorStop(1,"white");
ctx.fillStyle = grd; ctx.fillRect(10,10,150,80);

ctx.font = "30px Arial"; ctx.strokeText(“Smile",10,50);
</script>
```



Canvas 3D & WebGL

HTML5 <canvas> is also used for drawing 3D scene.
WebGL is based on OpenGL ES and provides an API for 3D graphics.
The programs consist of control code written in JavaScript and shader code which is executed on GPU.

```
// HTML5 element
<canvas id="myCanvas" width="600" height="400">
Your browser does not support the HTML5 canvas tag.
</canvas>

// Javascript
<script>
var c = document.getElementById("myCanvas"); var gl =
c.getContext("webgl");
renderScene(gl);
</script>
```

(http://carvisualizer.plus360degrees.com/threejs/)

Canvas libraries

- **PaperJS**

It provides clean programming interface to create and interact with vector graphics and bezier curves.

- **ProcessingJS**

It’s designed to write visualizations, images, and interactive content.

It allows browsers to display animations, visual applications, games and other graphical rich content.

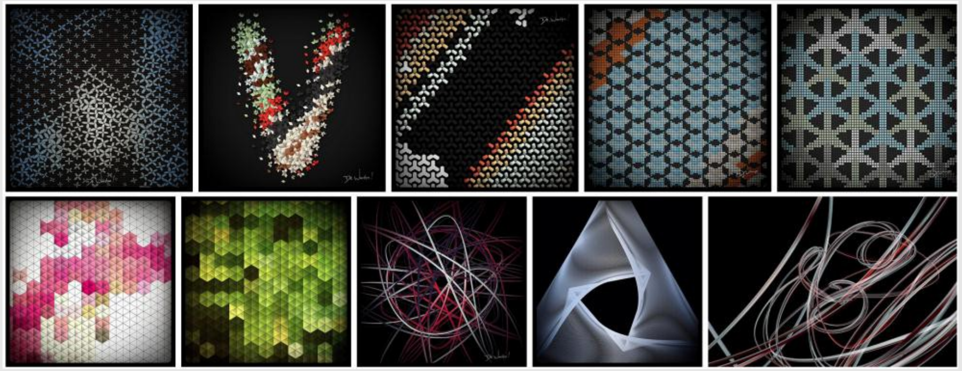
- **GoJS**

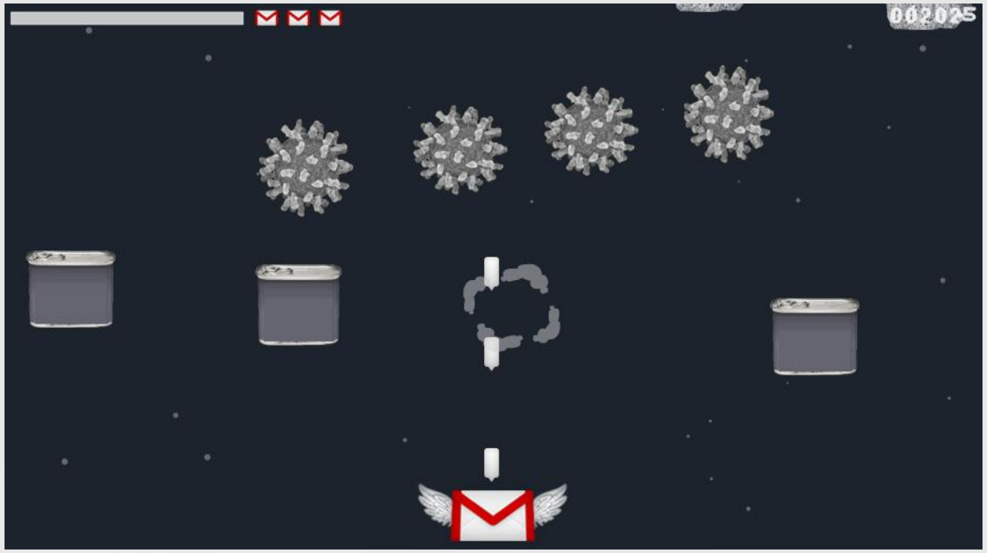
Construct interactive diagrams in easy way.

- **ThreeJS**

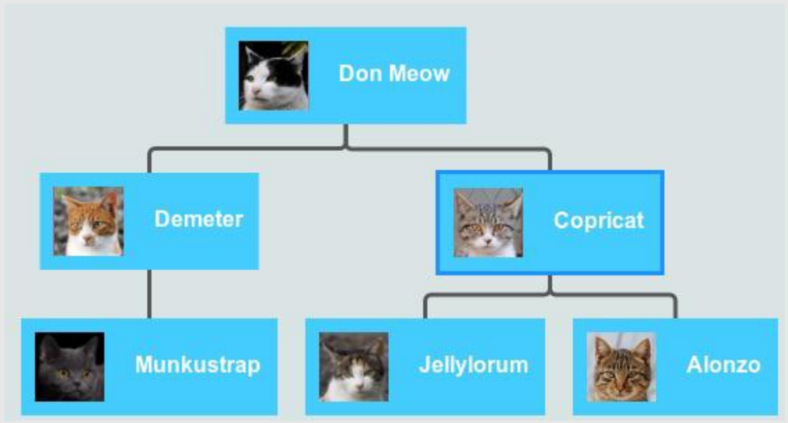
It’s a wrapper of WebGL allowing the creation of GPU-accelerated 3D animations running on browsers.

Canvas Examples


(PaperJS)



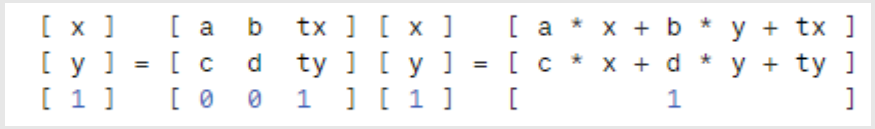
(ProcessingJS)



(GoJS)

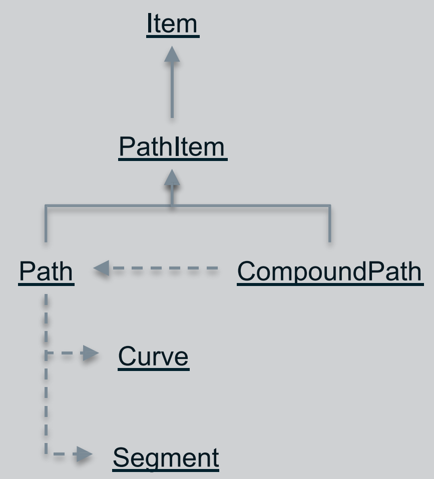
PaperJS reference

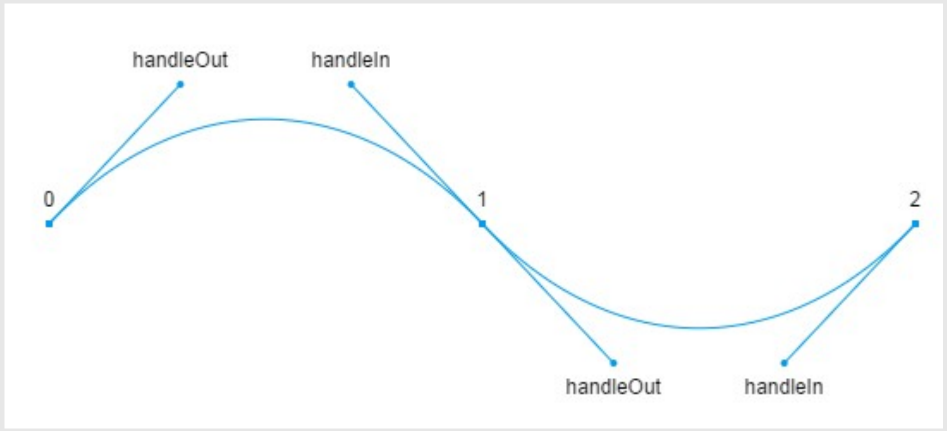
```
// Basic Types
Point(x, y)
Size(width, height) Rectangle(point, size)
Matrix(a, c, b, d, tx, ty)
```



```
[ x ]   [ a  b  tx ] [ x ]   [ a * x + b * y + tx ]
[ y ] = [ c  d  ty ] [ y ] = [ c * x + d * y + ty ]
[ 1 ]   [ 0  0  1  ] [ 1 ]   [         1          ]
```

(Affine transformation)





(path = curves + segments)

Collaboration

Operational Transformation (OT)

Operational Transformation (OT) is a class of algorithms that do multi-site real-time concurrency. OT is particularly suitable for implementing collaboration features such as group editing in the Web/Internet context. It has been adopted as a core technique behind the collaboration feature in Apache Wave and Google Docs.

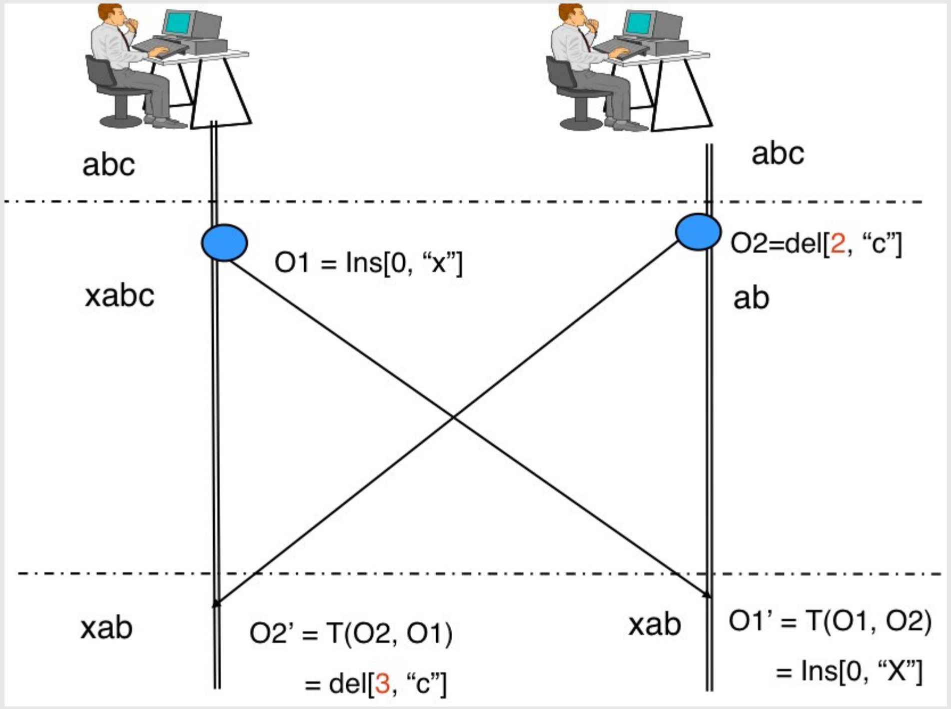
Collaborative systems using OT typically adopt a replicated architecture for the storage of shared documents to ensure good responsiveness in high latency environments, such as the Internet. The shared documents are replicated at the local storage of each collaborating site, so editing operations can be performed at local sites immediately and then propagated to remote sites. Remote editing operations arriving at a local site are typically transformed and then executed. The transformation ensures that application-dependent consistency criteria are achieved across all sites. The lock-free, nonblocking property of OT makes the local response time not sensitive to networking latencies

Operational Transformation (OT)

O1 = Insert[0, "x"]   // insert character "x" at position "0“
O2 = Delete[2, "c"]   // delete the character "c" at position "2“

⇩

O1` = T(O1, O2) = Insert[0, "x"]   // insert character "x" at position "0“
O2` = T(O2, O1) = Delete[3, "c"]   // delete character "c" at position “3“



ShareJS

ShareJS is an Operational Transformation (OT) library for browsers. It lets you easily do live concurrent editing in web application.
It’s very useful to let multiple users to view & edit the same data.
It can ensure eventual consistency between multiple users without retries, errors and any date being overwritten.

ShareJS generates operation for each edit. Operations are like git commit to document. If multiple users submit operations simultaneously, the server is responsible to transform the operations and apply to each client. Transformation is a bit like git rebase operation. The alogrithm is very careful to make sure every client ends up with the same document, no matter what order the operations are actually applied in.

ShareJS has functions defined for either plan text or JSON objects.

Example

```
// Client
sharejs.open( ‘GUID’ , 'json', function(err, doc) { if (doc.created) {
// initialize the shareJS document and submit to server for first time var op = {p:[ ], od: nulll, oi: { ‘items’ : [ ],
 ‘updates’ : [ ]}; doc.submitOp([op]);
} else {
// sync shareJS document from server var items = doc.snapshot.items;
…..     // do with items
var  updates = doc.snapshot.updates;
}
doc.on( ‘change’ , function(op) {
// do with op when other client submit operations
});
});
```

```
// server (Node.JS)
var http = require('http');
var app = express();   // express framework var sharejs = require('share'); Sharejs.server.attach(app, {db:
{type:'none'}}); http.createServer(app).listen(3000);
```

Real-time communication

Audio & Video basic

HTML5 introduces the <audio> and <video> elements to play audio & video media. That' s the standard way to play media without plugins.

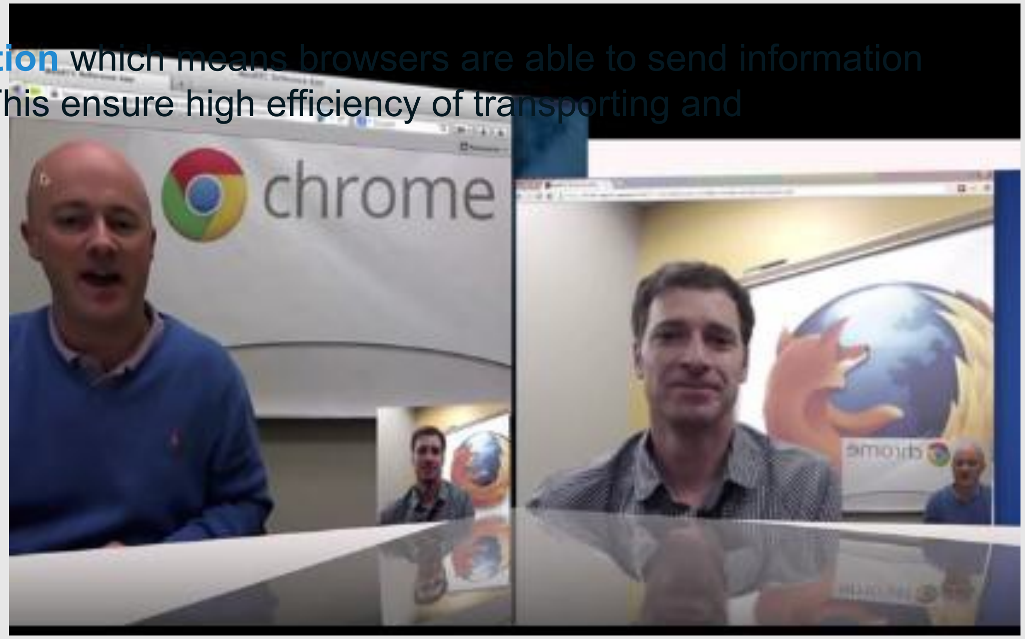
```
// HTML5 element
<audio autoplay controls>
  <source src="sample.mp3"/>
  <a> Download directly.</a>
</audio>
```

```
// HTML5 element
<video autoplay controls preload>
  <source src="sample.mp4"/>
  <p>The browser does not support video play</p>
</video>
```

WebRTC

WebRTC is free, open project that enables web browsers with Real-Time Communication(RTC) capabilities via simple JavaScript APIs (without plugins). The WebRTC components have been optimized to best serve this purpose. WebRTC W3C standards are under drafting.

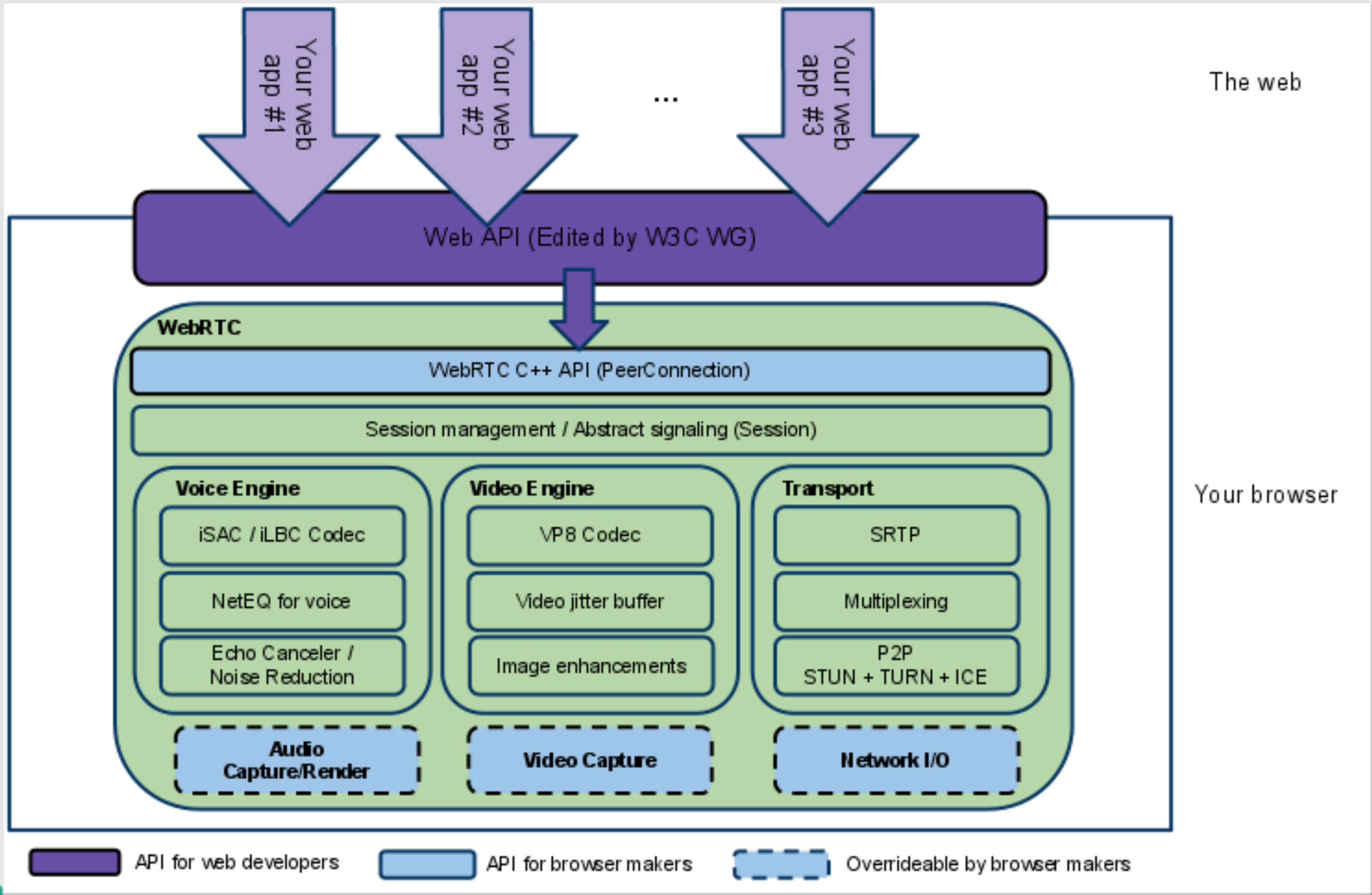
WebRTC allows **peer to peer communication** which means browsers are able to send information or stream without sending through server. This ensure high efficiency of transporting and dramatically reduce load at server side.



www.webrtc.org/demo

WebRTC Architecture

- Voice Engine
- Video Engine
- Transport



WebRTC API

- **getMediaStrea:**

get access to data streams, such as from the user's camera and microphone

- **RTCPeerConnection**

audio or video calling, with facilities for encryption and bandwidth management

- **RTCDataChannel**

peer-to-peer communication of generic data

WebRTC.io

WebRTC.io is a library providing an abstraction layer for WebRTC, aiming to simplify the HTML5 web standard WebRTC in a similar manner to socket.io with websockets.

```
// HTML5 at Client side
<video id="local" autoplay="autoplay"></video>
<video id="remote" autoplay="autoplay"></video>

<script src="/webrtc.io.js"></script>
<script> rtc.connect('ws://yourserveraddress:8001');
rtc.createStream({"video": true, "audio":false}, function(stream) {
rtc.attachStream(stream, 'local'); // get local stream
});
rtc.on('add remote stream', function(stream) { rtc.attachStream(stream, 'remote');   // get remote
stream
});
</script>
```

```
// Node.js at Server side require('webrtc.io').listen(8001);
```

Concurrent Note

Intent

In order to illustrate HTML5 & Web technologies are really powerful and easy to use, I want to build the HTML5 application “Concurrent Note” which allowing multiple users to collaboratively edit the same 2D document in real time. It also enable users chat in video. All the functionalities are built with pure HTML5.

What is it used for

- **Network meeting**

  Allow attendees to share the same document and video based communication.

- **Online education**

  It’s good for teacher to interact with students for specific course.

- **Collaborative work**

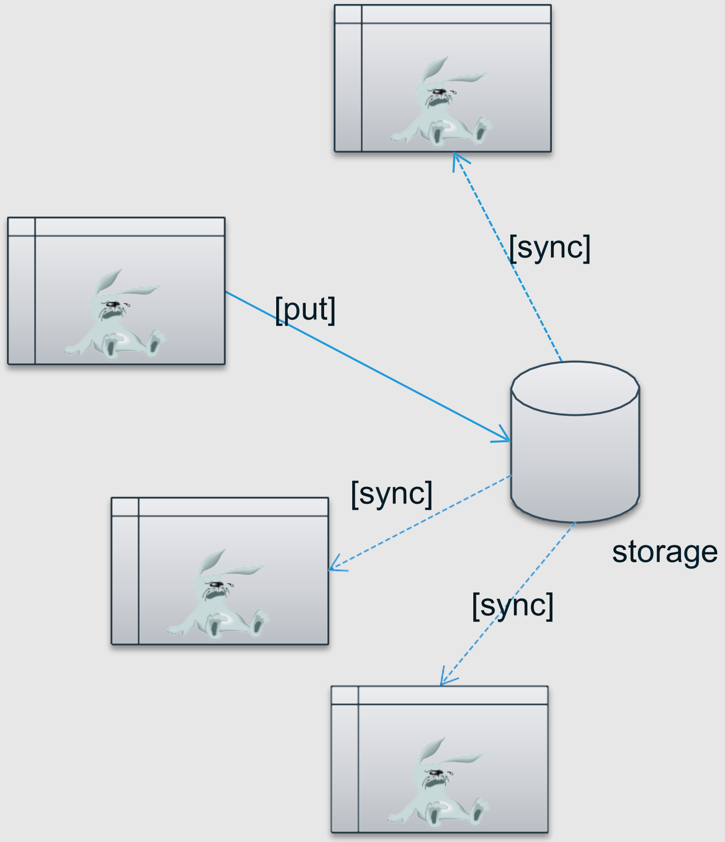
  Allow workmates / engineers to finish same job collaboratively.

- **Customization for Tizen**

  Maximize Tizen capabilities by customizing the application.

Collaborative Whiteboard

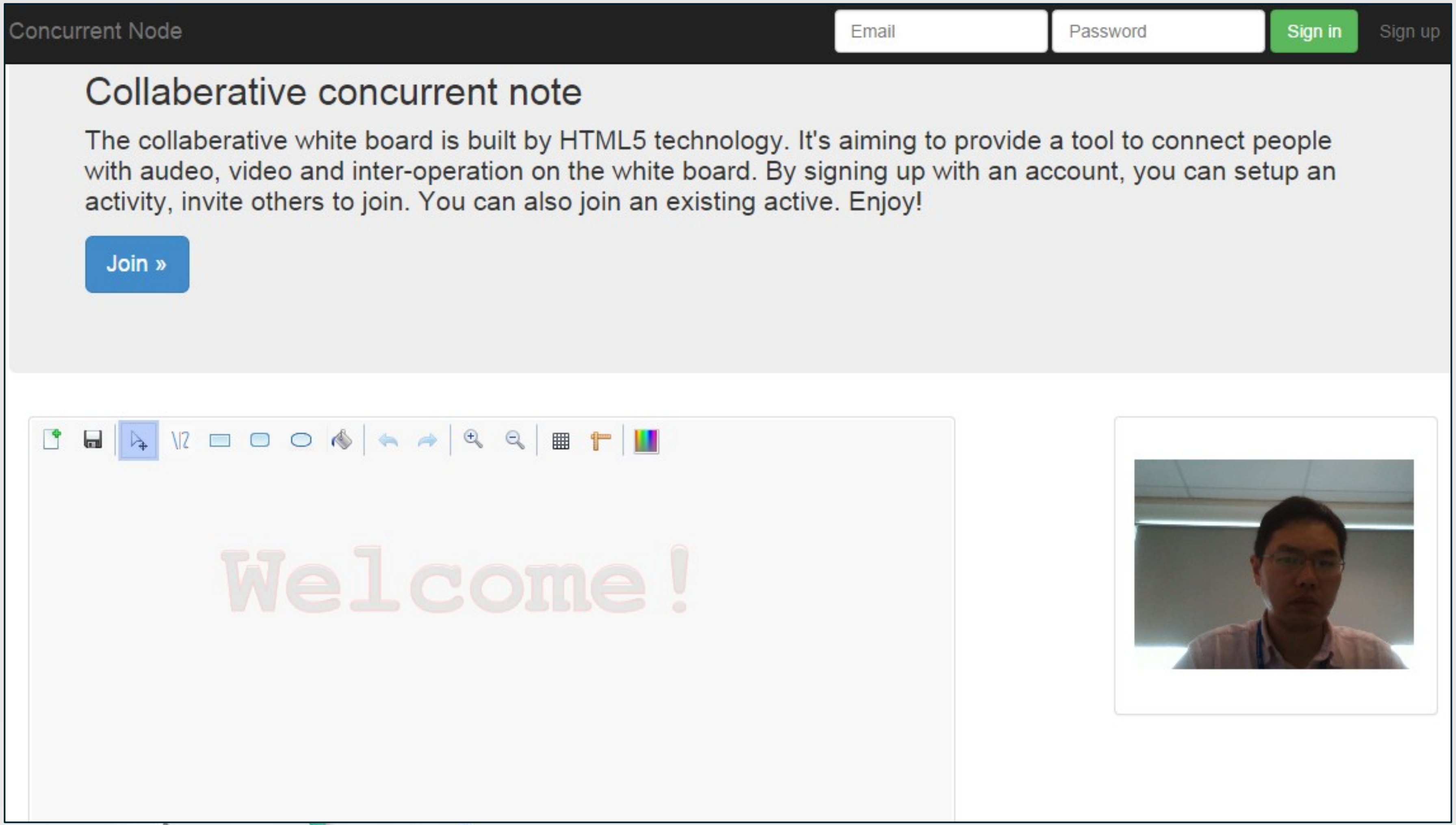
- **2D drawing capability**
  - Rich drawing features
  - Transaction & Storage
  - Assistance (snap, grip, ruler, group)
- **Broadcast delta state**
  - Client sends delta (minimal traffic load)
  - Server broadcasts the delta changes
  - Other clients synchronize to update local storage
- **Synchronize simultaneously**
  - Real-time synchronization
  - Keep data consistency without lock technique



Real-time communication

- **Video chatting**
  - WebRTC based video chatting
  - Full screen
  - Multiple users & multiple sessions
- **Screen sharing**

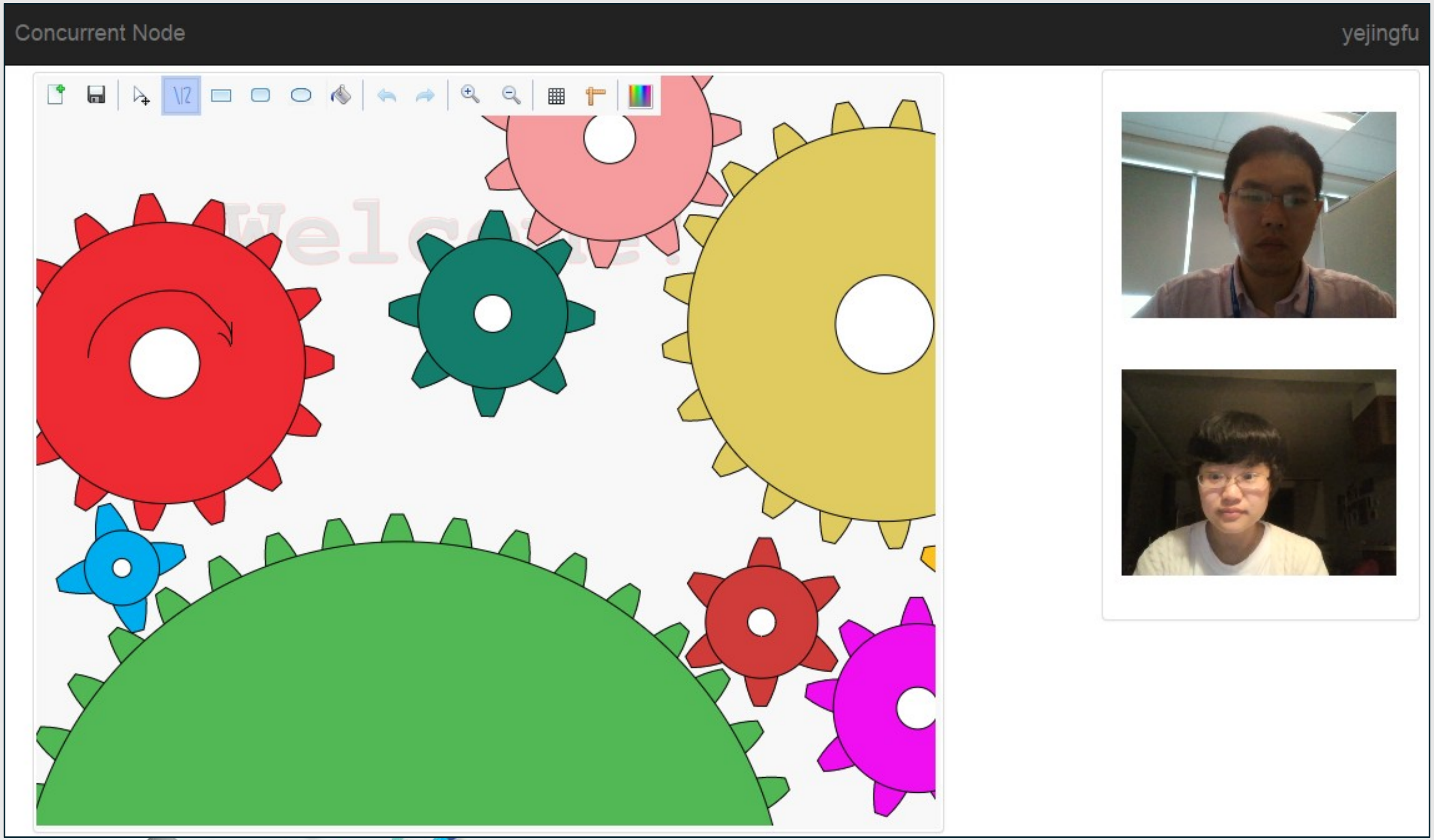
Application Snapshot



Collaberative concurrent note

The collaberative white board is built by HTML5 technology. It's aiming to provide a tool to connect people with audeo, video and inter-operation on the white board. By signing up with an account, you can setup an activity, invite others to join. You can also join an existing active. Enjoy!

Application Snapshot



Customization for Tizen

Easy to port to Tizen

This HTML5 application can easily port to Tizen without code change! That means you can use HTML5 APIs in Tizen web apps pretty much in the same way as you used in your web sites

What you need is to package the entire resources into a wgt file and install into Tizen device:

```
// config.xml
<widget …>
    <tizen:application id="Yejingfu.ConcurrentNote" package="Yejingfu" required_version="2.1"/>
    <content src="index.html"/>
    <icon src="icon.png"/>
    <name>Concurrent Note</name>
    <tizen:privilege name="http://tizen.org/privilege/tizen"/>
    <tizen:privilege name="http://tizen.org/privilege/filesystem.read"/>
    <tizen:privilege name="http://tizen.org/privilege/filesystem.write"/>
    <tizen:privilege name="http://tizen.org/privilege/wifi.read"/>
    <tizen:privilege name="http://tizen.org/privilege/notification" />
    <tizen:setting screen-orientation="portrait" encription="enable" install-location="auto"/>
</widget>
```

```
// package via zip command
$ zip -r concurrentnote.wgt config.xml *.html *.js *.css res/*
```

Customization for Tizen

- **Touch**

Touch capability make it easier to draw free-style curves on 2D canvas. The user interface becomes more friendly.

- **File system**

By using Tizen API to access local file system, the application can save / load the ShareJS document to local device. Besides it enable users to record the video chatting to local device as well.

- **Encryption**

Tizen provides encryption to run the application in secure mode.

- **Notification**

The Notification API provides a way to notify current user of events that happen in the application. Currently the application can notify when more users join this session. More cool notifications will be added in the future.

- **Location-based service**

Informing peer’s geographic location.

Examples

```
// Tizen notification
function onPeerJoin(name, location, profile) {
var ntf = new tizen.StatusNotification('Session', 'peer joind' { 'name': name, 'location':
location, 'profile': profile
});
tizen.notification.post(ntf);

setTimeout(function() { tizen.notification.remove(ntf.id);}, 3000);
}
```

```
// Tizen file system
function save(filepath, content) { var onSave = function(fileObj) {
fileObj.openStream('w', function(stream){ stream.write(content);
stream.close();
}.bind(file));
};
tizen.filesystem.resolve(path, onSave, function(err) {
// file not found
var pos = path.lastIndexOf('/'); var dir = path.substring(0, pos);
var filename = path.substring(pos + 1);
tizen.filesystem.resolve(dir, function(dirObj){onSave(dirObj.createFile(filename));});
});
}
```

Summary

HTML5 rocks

HTML5 is a collection of new technologies, unified under an official standard specification, to make the Internet cleaner, leaner, faster, and way more visually appealing.

It makes the job easier for developers, allowing them to produce stellar web applications for all devices. More and more developers chose HTML5 for building mobile apps. Both mobile and desktop are expanding support for HTML5.

HTML5 would be key to business apps for company.

If you want to learn more HTML5 very cool features and demonstrations, please visit: http://www.html5rocks.com

HTML5 consolidate Tizen

HTML5 provides easier way to develop applications for Tizen.

Tizen differentiates itself from iOS and Android by allowing developers to code in HTML5, JavaScript and CSS rather than Object-C and Java. This would attract developers with shorter development cycles and lower costs by avoiding experiences of native code.

HTML5 would be the first alternative to develop app for Tizen.

TIZEN™

DEVELOPER SUMMIT

2014

SHANGHAI

fil!:ffi}e.rat9fil!:mle.rat9fil!:ml

TIZEN7f *Md$ti (.t 5$)*