



体验丰富的 Tizen HTML5 应用

Jingfu.Ye@intel.com



概述

- 背景介绍
- 图形功能
- 协作
- 实时通信
- 并发备注
- **Tizen** 定制
- 总结

背景介绍



HTML5 新特性

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](#)

HTML



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

使用 HTML5 的优势

- **零安装和及时升级**

它能够增强应用的可访问性和可行性，而且更新能够在后台进行并即时生效

- **离线缓存**

支持应用在未连接互联网的情况下运行。

- **图形、音频和视频**

借助丰富的炫酷特性，以标准方式增强应用，无需插件。

- **跨平台和跨浏览器**

一次编写，随处运行！

- **移动设备上的混合应用**

将 HTML5 用于移动开发，实现双赢。

- **Websocket**

支持客户端与服务器进行全双工通信。

图形功能



Canvas 简介

HTML5 具有 `<canvas>` 标记，其内容可使用 JavaScript 进行渲染。为了利用 HTML5 canvas，我们需要将 `<canvas>` 标记置于 HTML 文档内。然后，我们可以在 JavaScript 中访问 canvas，并利用 canvas API 绘制可视化图形。

随着更多的浏览器可为 HTML5 canvas 标记提供硬件加速功能，使用 JavaScript 移动构建流畅、复杂的动画将变得更为容易。而且更多的开发工具将不断涌现，进一步简化 canvas 动画的创建过程。

Canvas 元素具有 canvas 上下文，这种对象具有的属性和方法可供您用于在 canvas 元素内渲染图形。

Canvas 上下文可以是 2D 和 3D(WebGL)。

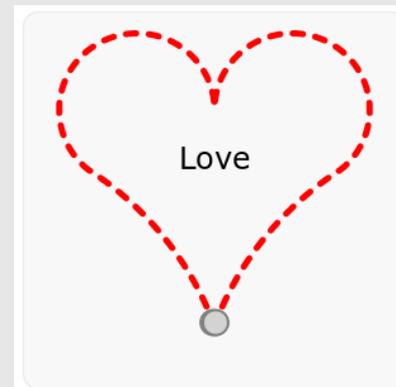
Canvas 2D

HTML5 `<canvas>` 元素用于在网页上绘制 2D 图形。Canvas 提供路径、方框、圆圈、文本和图像的绘制方法。

```
// 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 和 WebGL

HTML5 `<canvas>` 还可用于绘制 3D 场景。
WebGL 基于 OpenGL ES，可为 3D 图形提供 API。
这两种程序包含用 JavaScript 编写的控制代码和在 GPU 上执行的着色器代码。

```
// 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>
```



[\(http://carvisualizer.plus360degrees.com/threejs/\)](http://carvisualizer.plus360degrees.com/threejs/)

Canvas 库

- **PaperJS**

它可提供整洁的编程接口创建矢量图形和贝塞尔曲线，并与它们进行交互。

- **ProcessingJS**

它设计用于编写可视化图形、图像和交互式内容，支持浏览器显示动画、可视化应用、游戏和其他包含丰富图形的内容。

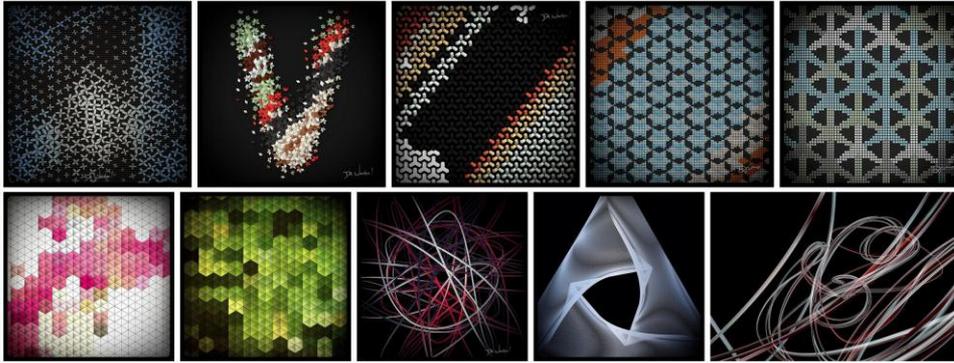
- **GoJS**

轻松构建交互式图表。

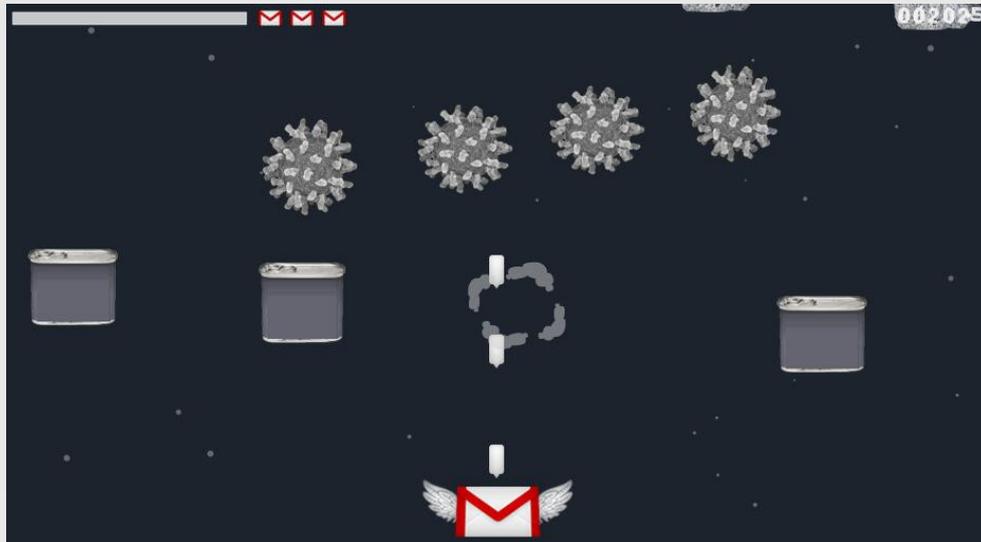
- **ThreeJS**

它是 WebGL 的封装程序，支持创建可在浏览器上运行的 GPU 加速 3D 动画。

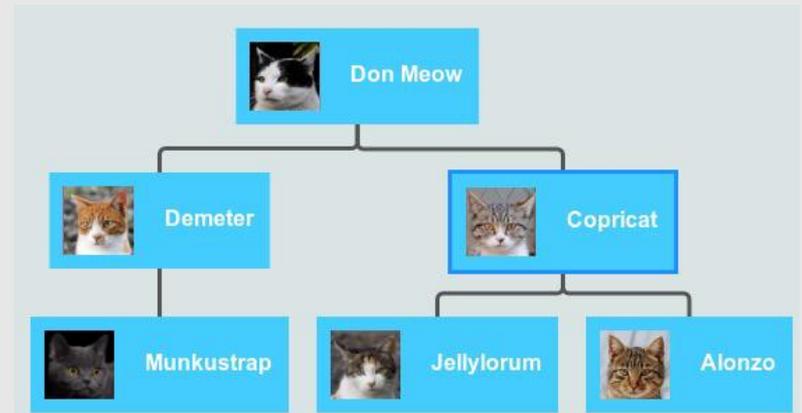
Canvas 示例



(PaperJS)



(ProcessingJS)



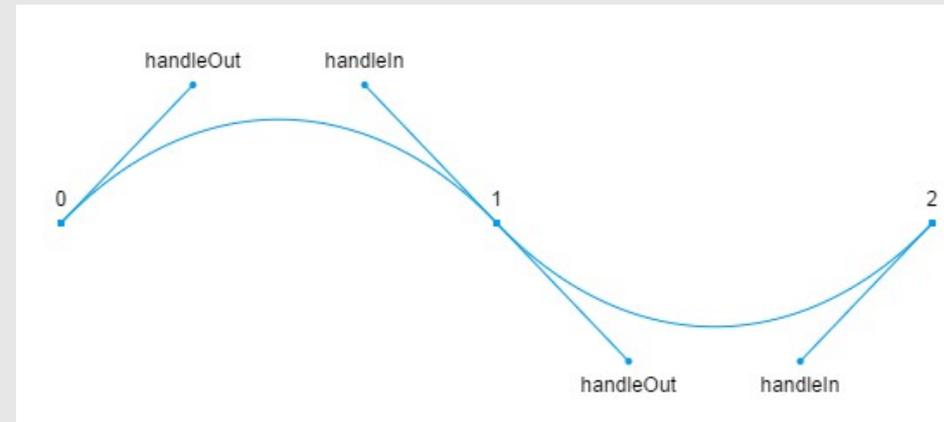
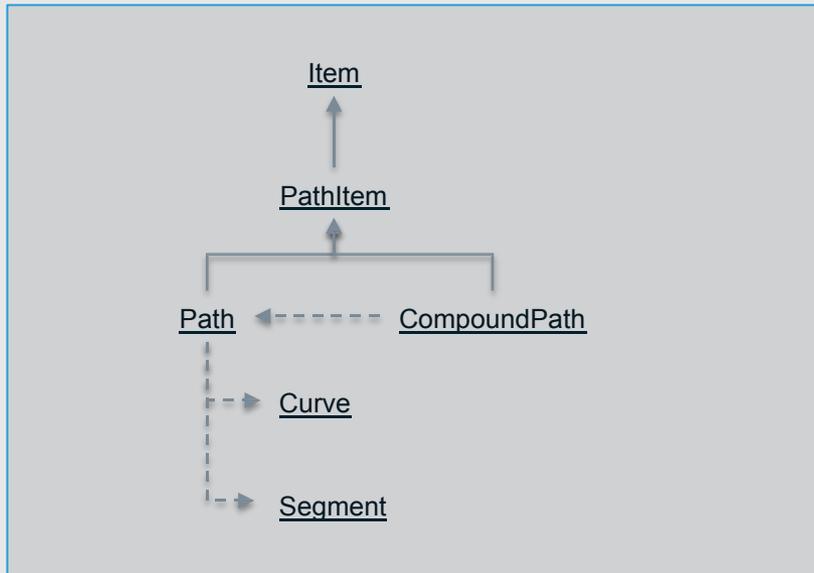
(GoJS)

PaperJS 参考

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

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & tx \\ c & d & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a * x + b * y + tx \\ c * x + d * y + ty \\ 1 \end{bmatrix}$$

(仿射变换)



(路径 = 曲线 + 片段)

协作



操作转换 (OT)

操作转换 (OT) 是一类可实施多站点实时并发性的算法。OT 尤其适合在 Web/互上下文中实施组编辑 (group editing) 等协作特性，已被用作 Apache Wave 和 Google Docs 中协作特性的核心技术。

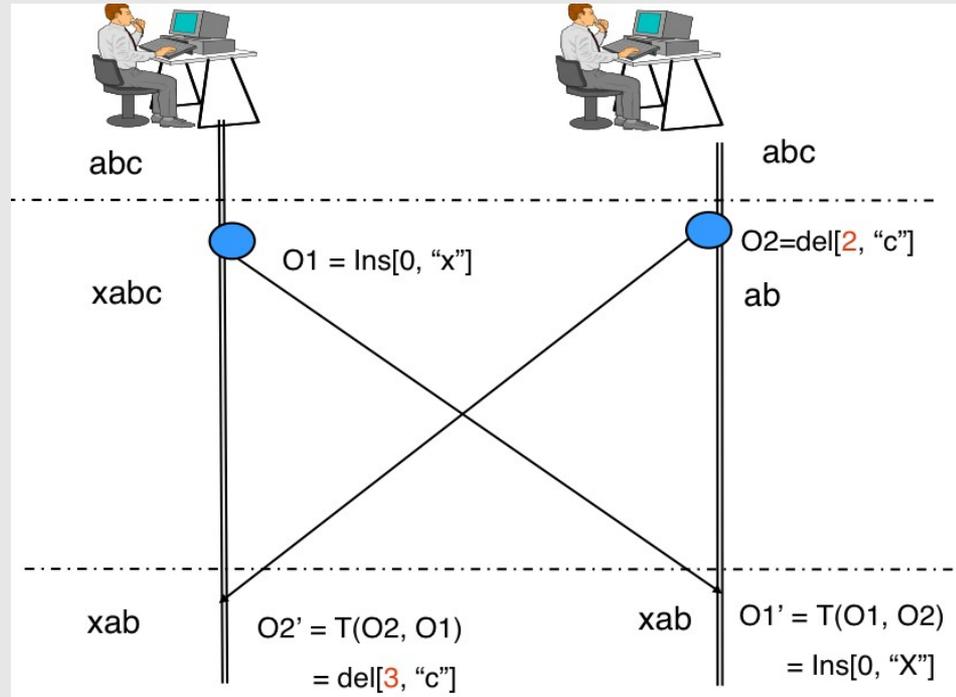
使用 OT 的协作系统通常采用一种复制架构 (replicated architecture) 存储共享文档，以确保在互联网等高延迟环境中实现出色的响应性能。共享文档可复制到每个协作站点的本地存储中，以便编辑操作可在本地站点即时实施，然后传送到远程站点。到达本地站点的远程编辑操作在执行之前通常需要进行转换。通过转换，可确保在所有站点之间实现依赖应用的一致性标准。OT 的无锁非阻塞属性使得本地响应时间不受网络延迟的影响。

操作转换 (OT)

$O1 = \text{Insert}[0, "x"]$ // insert character "x" at position "0"
 $O2 = \text{Delete}[2, "c"]$ // delete the character "c" at position "2"



$O1' = T(O1, O2) = \text{Insert}[0, "x"]$ // insert character "x" at position "0"
 $O2' = T(O2, O1) = \text{Delete}[3, "c"]$ // delete character "c" at position "3"



ShareJS

ShareJS 是一种面向浏览器的操作转换 (OT) 库，支持用户在 web 应用中轻松进行实时并发编辑。

让多个用户查看和编辑相同数据能够带来诸多裨益，可确保在多个用户之间实现最终一致性，同时避免重试、错误和覆盖任何日期。

ShareJS 可为每个编辑指令生成操作。操作如同文档的 `git commit`。如果多个用户同时提交操作，服务器负责转换操作并将其应用于每个客户端。转换有点像 `git rebase` 操作。该算法会全力确保每个客户端最后使用相同文档，无论操作的实际应用顺序如何。

ShareJS 具有为纯文本或 JSON 对象定义的功能。

示例

```
// 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: null, 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);
```

实时通信



音频和视频基本元素

HTML5 采用 `<audio>` 和 `<video>` 元素播放音频和视频媒体，这是不用插件播放媒体的标准方式。

```
// 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>
```

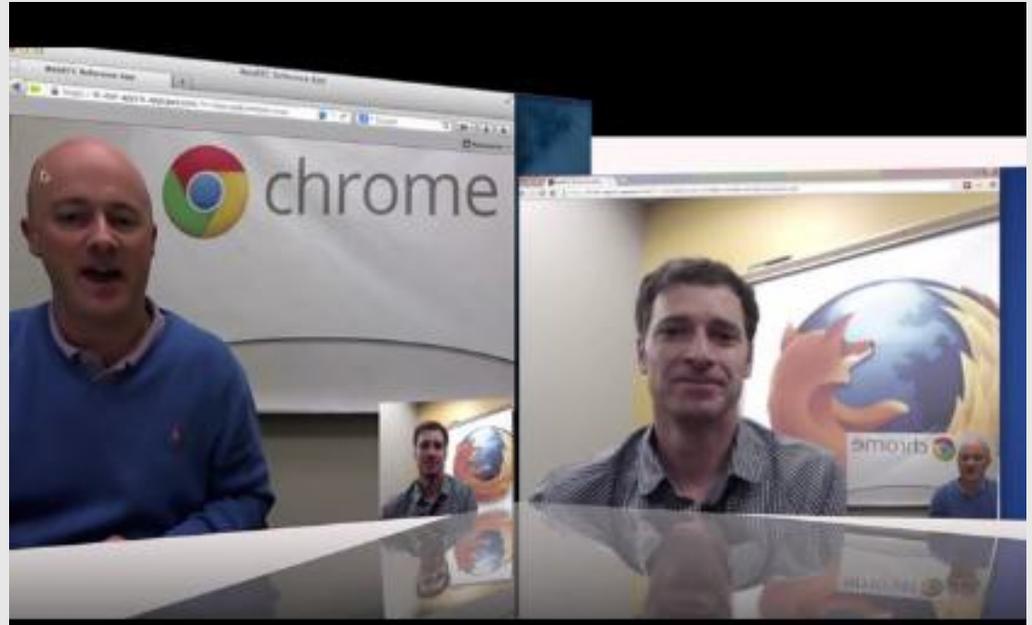


WebRTC

WebRTC 是一种免费的开源项目，可通过简单的 JavaScript API（不带插件）为 web 浏览器提供实时通信 (RTC) 功能。为完全实现这一目的，WebRTC 组件进行了优化。

WebRTC W3C 标准正在起草中。

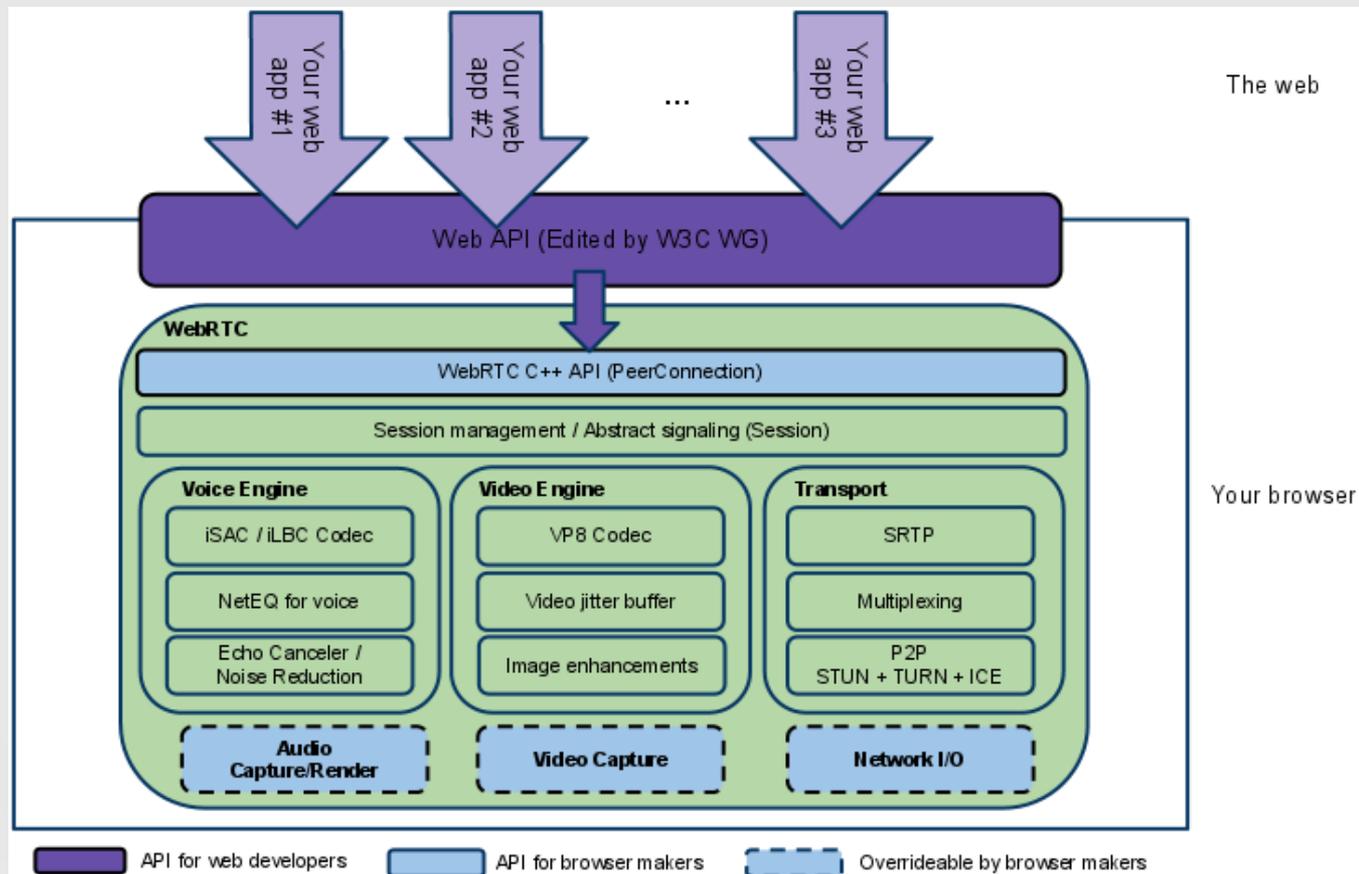
WebRTC 支持**对等通信**，这意味着浏览器能够发送信息或流媒体，无需通过服务器发送。这能够确保高效传输，显著减少服务器端的负载。



www.webrtc.org/demo

WebRTC 架构

- 语音引擎
- 视频引擎
- 传输



WebRTC API

- **getMediaStream:**
访问数据流，如通过用户的摄像头和麦克风
- **RTCPeerConnection**
借助加密和带宽管理工具进行音频或视频通话
- **RTCDataChannel**
通用数据的对等通信

WebRTC.io

WebRTC.io 是一种可为 WebRTC 提供抽象层的库，旨在以类似于使用 websocket 的 socket.io 的方式，简化 HTML5 web 标准 WebRTC。

```
// 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);
```

并发备注



目的

为了说明 HTML5 和 Web 技术极为强大且易于使用，我想要构建 HTML5 应用的“并发备注”，以支持多个用户实时协作编辑相同的 2D 文档。它还支持用户视频聊天。所有这些功能可使用纯 HTML5 进行构建。

它有何用处

- **网络会议**

支持参加者使用相同的文档和视频进行通信。

- **在线教育**

它可有效帮助教师和学生就特定课程进行互动。

- **协同工作**

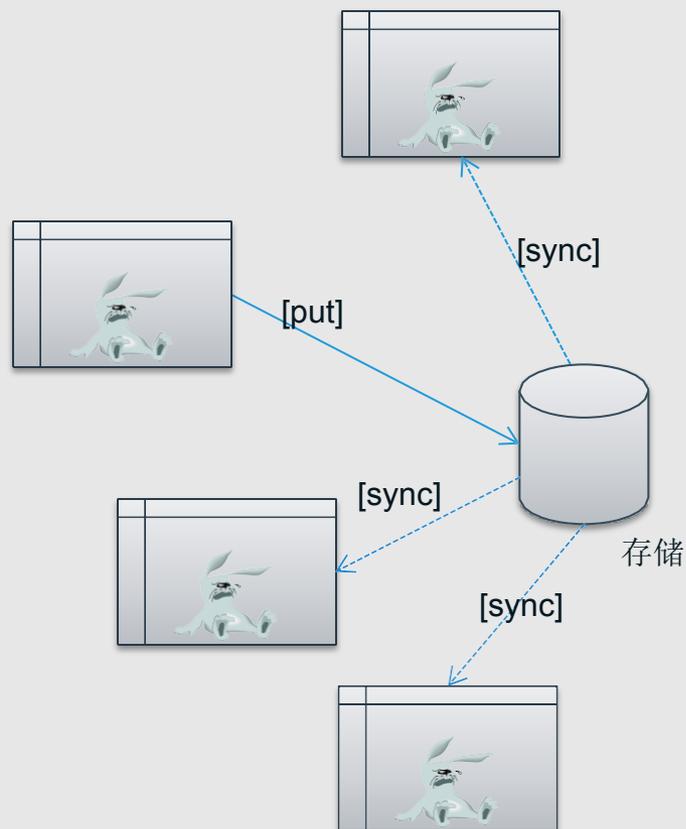
支持同事/工程师协同完成相同工作。

- **Tizen 定制**

通过量身定制应用，充分利用 Tizen 功能。

协作式白板

- **2D 绘图功能**
 - 丰富的绘图特性
 - 交易和存储
 - 协助（快照、紧握、标尺、群组）
- **广播 delta 状态**
 - 客户端发送 **delta**（最低流量负载）
 - 服务器广播 **delta** 变化
 - 其他客户的可同步更新本地存储
- **同时同步**
 - 实时同步
 - 在不用锁定技术的情况下保持数据一致性



实时通信

- 视频聊天
 - 基于 WebRTC 的视频聊天
 - 全屏
 - 多个用户和多个会话
- 屏幕共享

应用快照

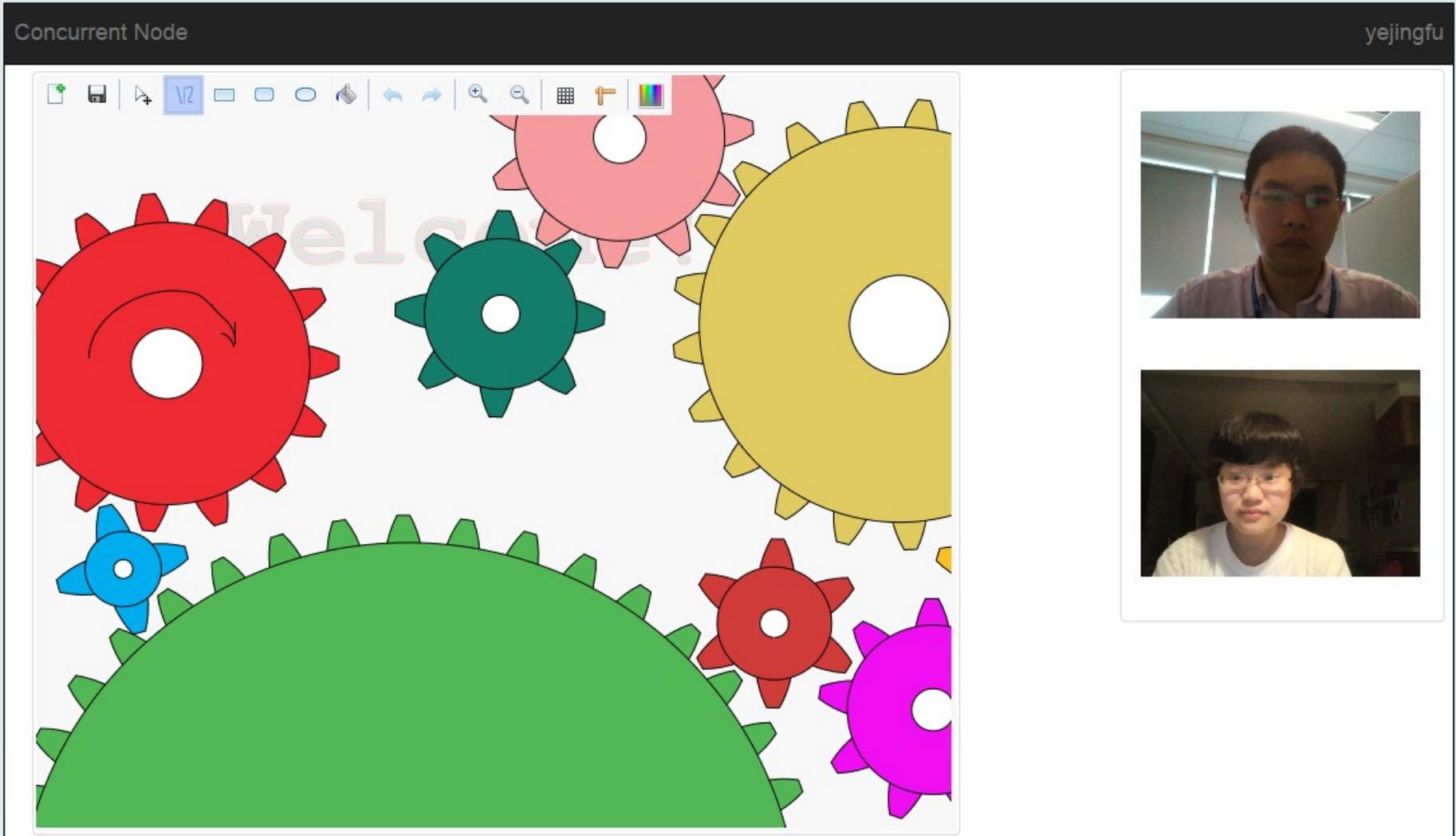
Concurrent Node [Sign up](#)

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!



应用快照



Tizen 定制



轻松移植到 Tizen

这种 HTML5 应用无需修改代码便可轻松移植到 Tizen! 这意味着您能够基本像在网站中一样, 在 Tizen web 应用中使用 HTML5 API。

您只需要将全部资源打包到一个 wgt 文件并安装于 Tizen 设备中:

```
// 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"/>
<widget ...>
<content src="index.html"/>
<name>Concurrent Note</name>
<tizen:privilege
```

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

Tizen 定制

- 触控

触控功能可帮助用户在 2D canvas 上轻松绘制自由式曲线 (free-style curve)。这种用户界面更加易用。

- 文件系统

通过使用 Tizen API 访问本地文件系统，应用能够保存/加载 ShareJS 文档至本地设备。此外，它还支持用户将视频聊天内容记录在本地设备中。

- 加密

- 触控

触控功能可帮助用户在

2D canvas 上轻松绘制自由式曲线 (free-style curve)。

户界面更加易用。

这种用

通过文件系统

API 访问本地文件系统，应用能够保存/

加载 ShareJS

示例

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

总结



HTML5 功能强大

HTML5 包含一系列符合统一的正式标准规范的新技术，能够使互联网更简洁、精简和快速，以及具有更强的视觉吸引力。

它能够简化开发人员的工作，帮助他们为所有设备开发出出色的 web 应用。越来越多的开发人员选用 HTML5 来构建移动应用。移动应用和桌面应用正在扩展对 HTML5 的支持。

对于公司的业务应用，HTML5 必不可少。

如欲了解 HTML5 的更多炫酷特性和演示，请访问：<http://www.html5rocks.com>

HTML5 增强 Tizen

HTML5 有助于更轻松地为 Tizen 开发应用。

Tizen 与 iOS 和 Android 的不同之处在于，它支持开发人员使用 HTML5、JavaScript 和 CSS（而非 Object-C 和 Java）进行编码。这能够使用本地代码，从而吸引开发周期较短且预算较少的开发人员。

HTML5 是为 Tizen 开发应用的首要选择。



TIZEN™

开发人员
峰会
2014



SHAN GHAI

fill:ffj}e.rat9fil!:mle.rat9fil!:ml

TIZEN7f :M d\$ti

(..t5\$)