
Google™
10

Google Wave Media APIs:
Attachments Can Surf Too!

Seth Covitz, Jimin Li & Phil Liao
May 20th, 2010

View live notes and ask questions about this session on Google Wave

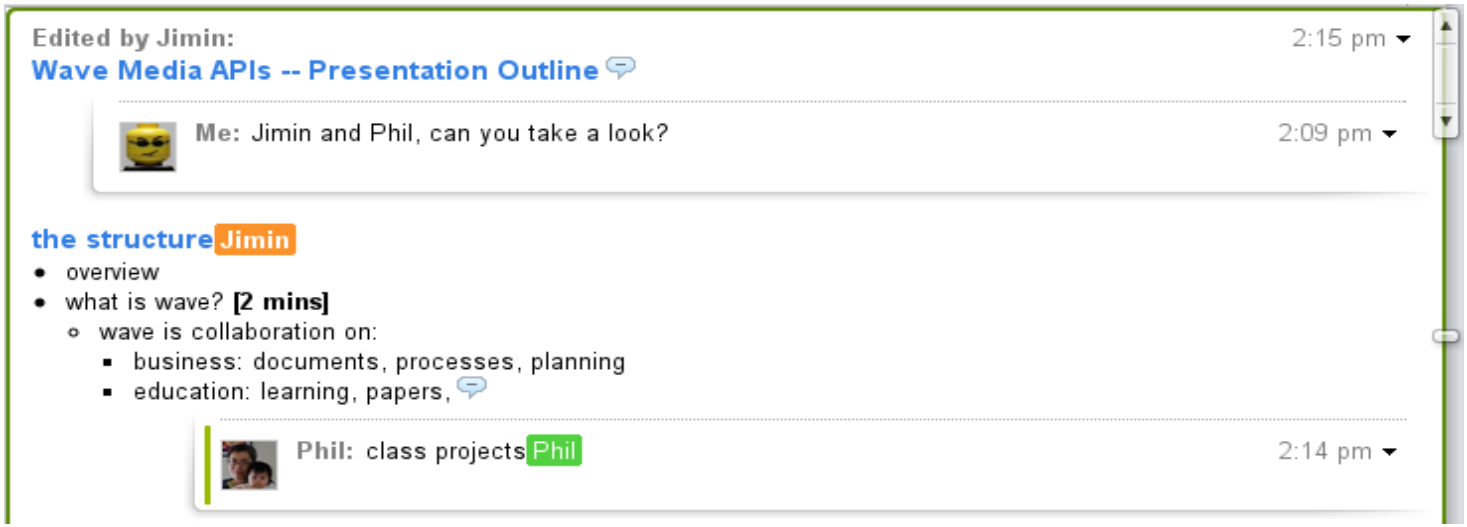
http://bit.ly/wave-media-api

Introducing the Wave Media API

Google 10

What is Google Wave?

- Wave is collaboration on:
  - **business**
    - documents, processes, meetings
  - **education**
    - learning, research, projects
  - **consumer**
    - photo albums, party planning, community groups, hobbies
- Wave is communication that streamlines your collaboration.
- Wave is live and real-time.



Google I/O 10

What is rich content?

- Any external content added to a wave.
  - photos, music, video
  - documents, diagrams, charts, tables
  - custom file types: text, xml, csv

Google I/O 10

With rich content, users want to ...

- import it
- view it
- play it
- share it
- edit it
- convert it
- export it
- publish it
- synchronize it
- and more...

Google I/O 10

Introducing the Wave Media APIs

Full access to attachments:

- robots can be used to:
  - create, fetch and delete attachments
- gadgets can be used to:
  - view documents
  - play media
- extensions can be used to share:
  - importers, exporters and viewers

Wave Robot Attachment APIs

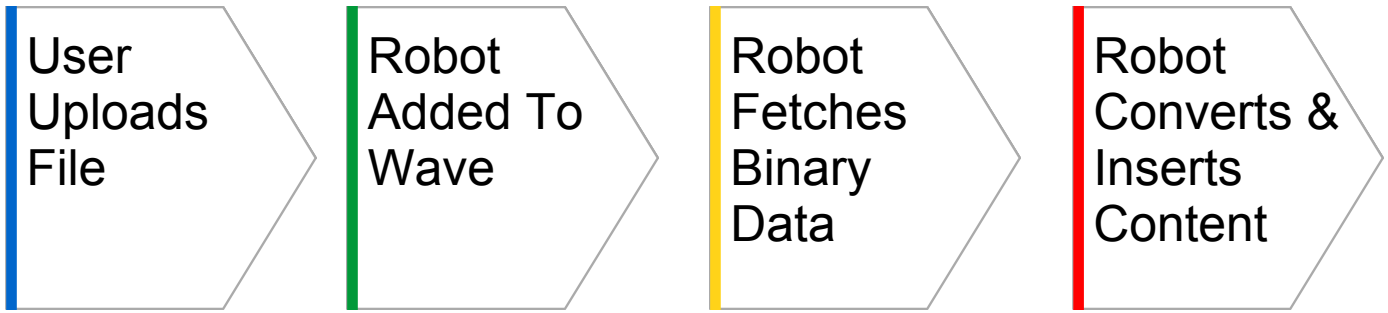
Wave Attachment Read API

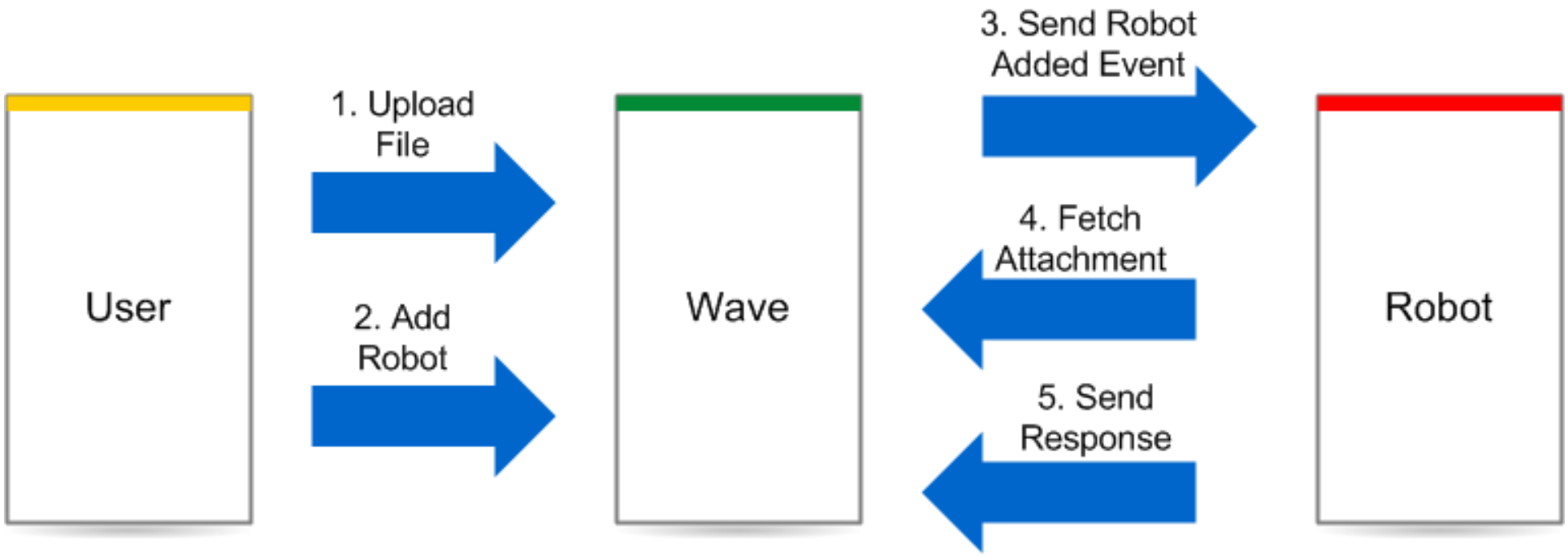
Retrieve attachment information

- Fetch attachment content
- Get attachment attributes:
  - Id
  - Caption
  - URL
  - MIME Type

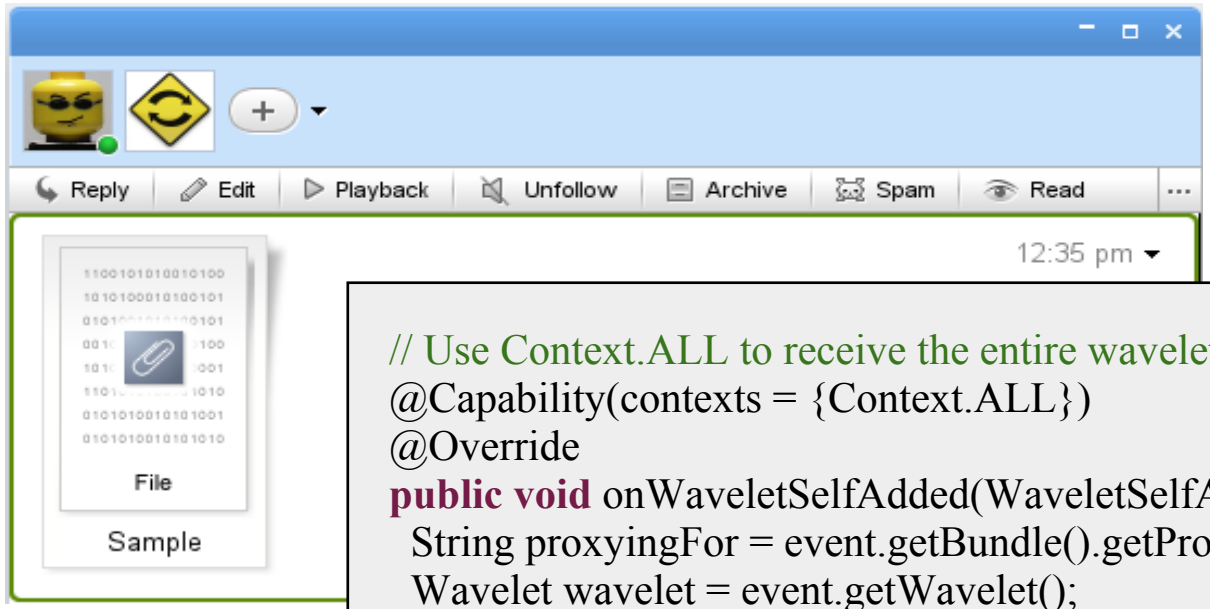
Google I/O 10

Import: Flow


User Uploads File
Robot Added To Wave
Robot Fetches Binary Data
Robot Converts & Inserts Content


User
1. Upload File
2. Add Robot
Wave
3. Send Robot Added Event
4. Fetch Attachment
5. Send Response
Robot

Import: how does it work?




```
// Use Context.ALL to receive the entire wavelet.
@Capability(contexts = {Context.ALL})
@Override
public void onWaveletSelfAdded(WaveletSelfAddedEvent event) {
  String proxyingFor = event.getBundle().getProxyingFor();
  Wavelet wavelet = event.getWavelet();
  Blip blip = event.getBlip();

  // Use proxyingFor to allow one robot to act as many.
  if (proxyingFor.equals("import-txt")) {
    importText(blip);
  } else if (proxyingFor.equals("export-txt")) {
    exportText(wavelet);
  }
}
```

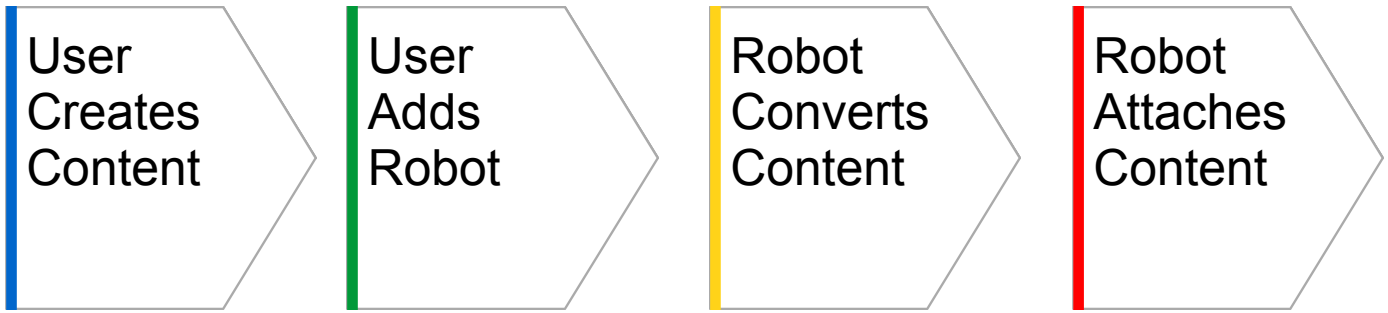
Import: how does it work?

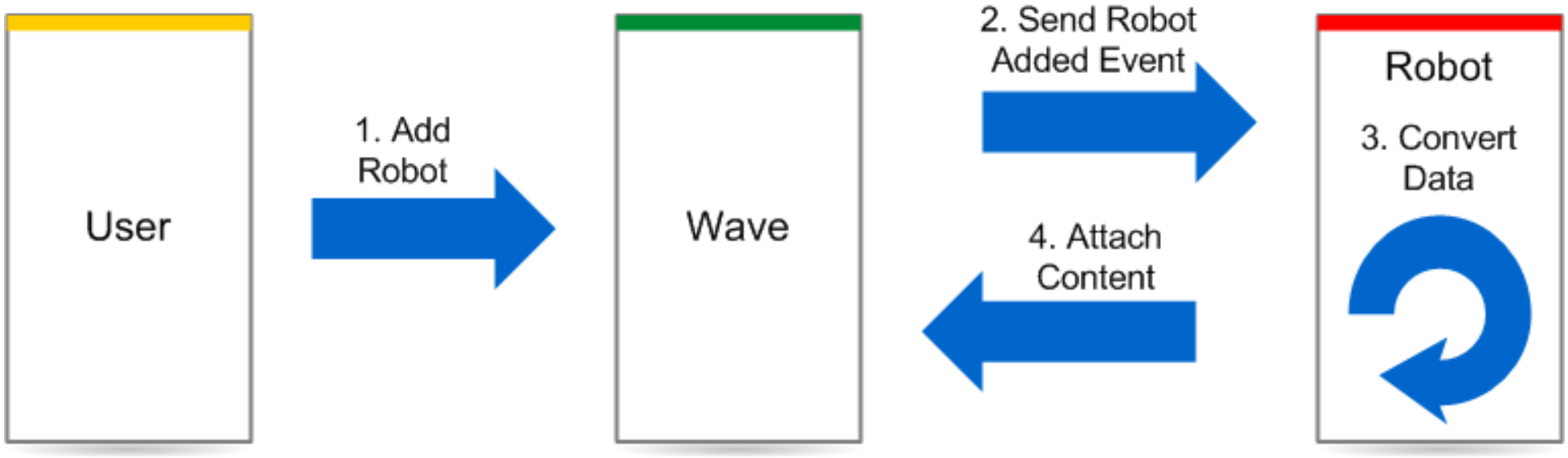
```
private void importText(Blip blip){

  BlipContentRefs attRef = blip.first( ElementType.ATTACHMENT);

  if (attRef != null) {
    Attachment attachment = (Attachment) attRef.value();
    // Replace the attachment with its imported content.
    attRef.replace(new String(attachment.getData()));
  }
}
```

Wave Attachment Write API

- Create an attachment with:
  - Caption
  - Attachment data in byte array
- Delete an attachment by:
  - removing the corresponding element
- Update the attachment by:
  - replacing it with a new attachment

Export: Flow


User Creates Content
User Adds Robot
Robot Converts Content
Robot Attaches Content


User
1. Add Robot
Wave
2. Send Robot Added Event
4. Attach Content
Robot
3. Convert Data

Export: how does it work?

```
private void exportText(Wavelet wavelet) {
  StringBuilder exportedWave = new StringBuilder();

  // Iterate over blips in the wavelet.
  for (Blip blip : wavelet.getBlips().values()) {
    // Iterate over the content in the blip.
    exportedWave.append(blip.getContent());
  }

  Attachment attachment = new Attachment(
      "exported!",
      exportedWave.toString().getBytes());
  wavelet.reply("\n").append(attachment);
}
```

Google I/O 10

Demo - Origami Photo Album

What does it do?

- Composes two different styles of pictures

How does it work?

- Gets all the image attachment URLs
- Fetches URL contents, resizes the images, and composes the images into two styles
- Uploads images

Demo - Bar Chart

What does it do?

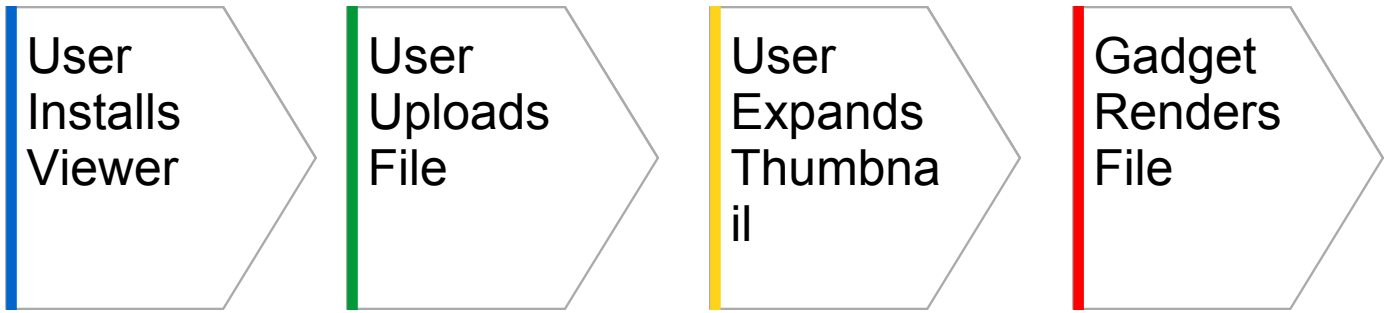
- Creates bar chart based on text file data

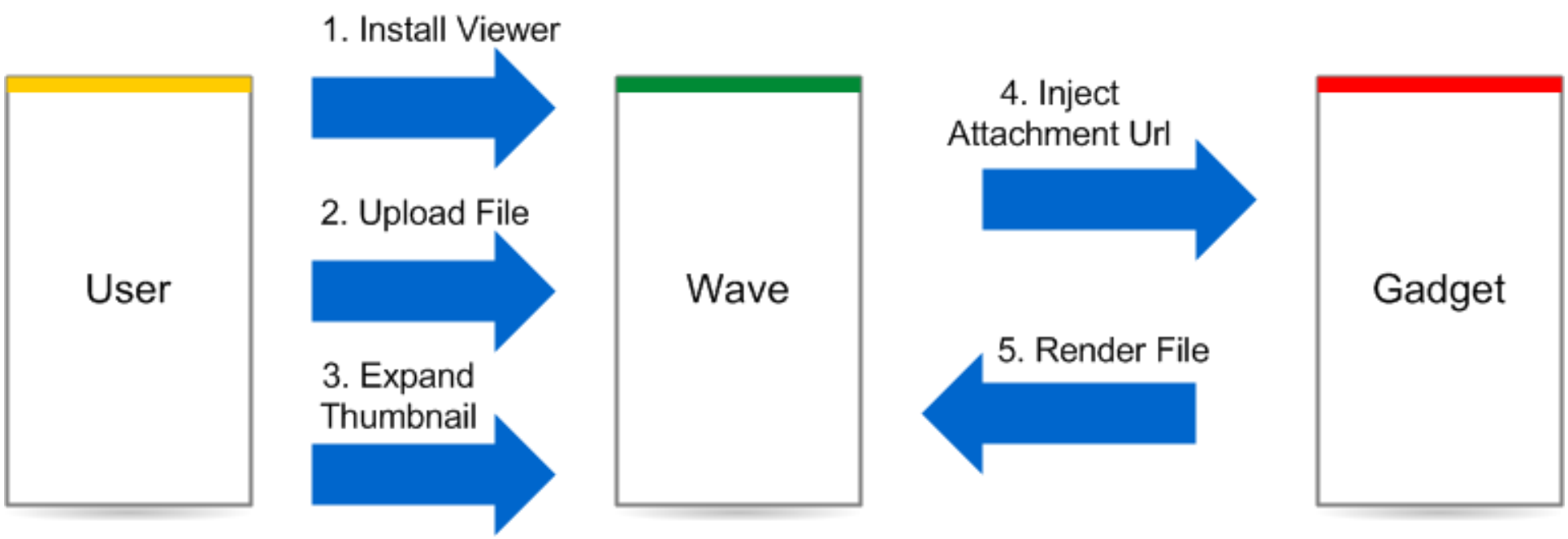
How does it work?

- Reads attachment data
- Creates bar chart

Gadget Viewer API

View: Flow


User Installs Viewer
User Uploads File
User Expands Thumbna il
Gadget Renders File

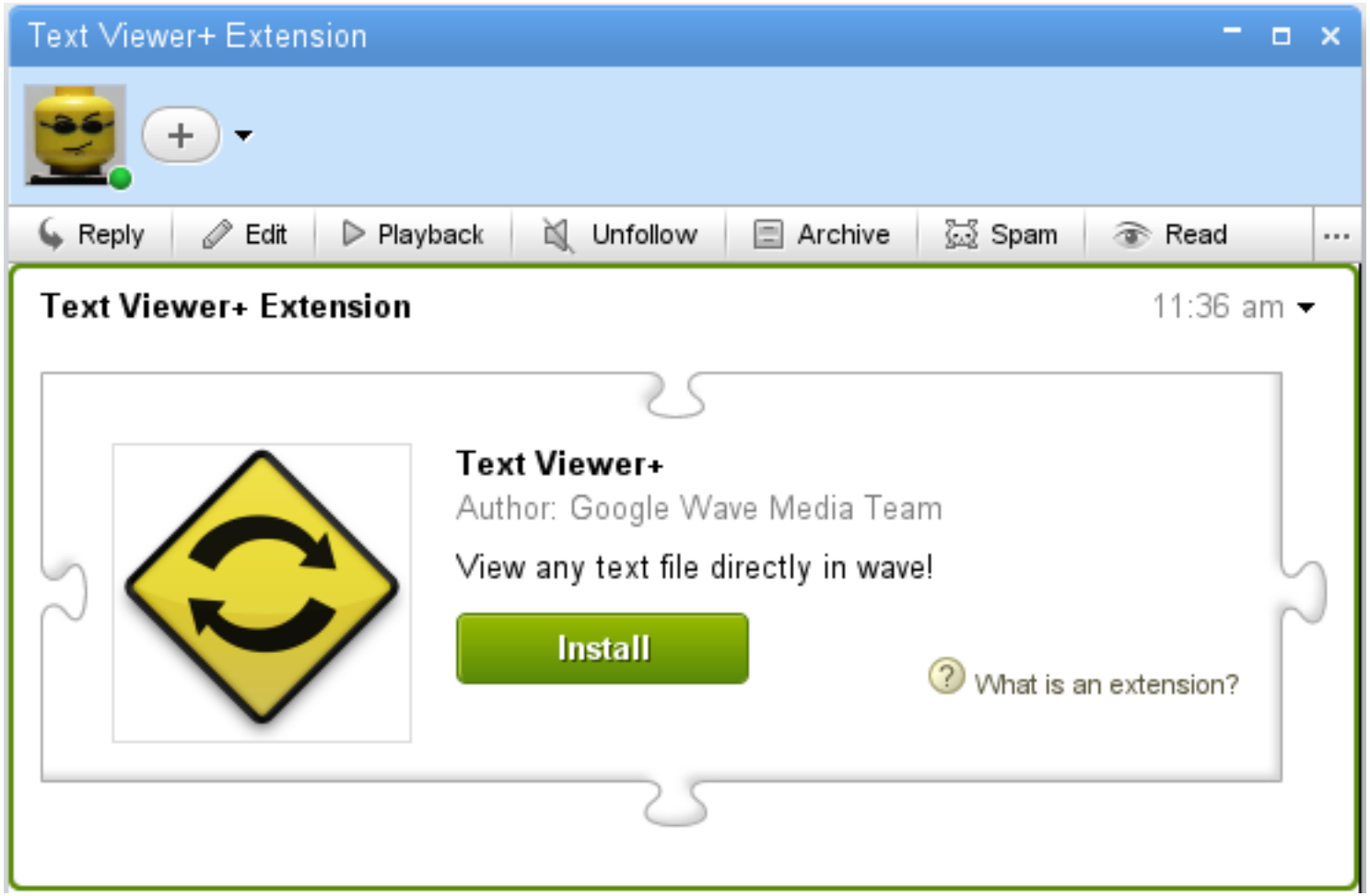

1. Install Viewer
2. Upload File
3. Expand Thumbnail
User
Wave
4. Inject Attachment Url
5. Render File
Gadget

View: how does it work?

```
<script type="text/javascript">
var div = document.getElementById('content_div');

function init() {
  if (wave && wave.isInWaveContainer()) {
    wave.setStateCallback(onStateUpdated);
  }
}

function onStateUpdated() {
  var attachmentUrl =
      wave.getState().get('attachment_url');
  gadgets.io.makeRequest(attachmentUrl, onUrlFetched);
}

function onUrlFetched(obj) {
  div.innerHTML = obj.text;
  gadgets.window.adjustHeight();
}

gadgets.util.registerOnLoadHandler(init);
</script>
```

Media Extensions

Media Extensions



Media Extensions

```
<extension
    name="Text Viewer+"
    description="View any text file directly in wave!"
    thumbnailUrl="http://media-api.appspot.com/.../yellow-sign.png">
  <author name="Google Wave Media Team"/>
  <attachmentHook
    name="Text Viewer"
    url="http://media-api.appspot.com/.../text-viewer.xml"/>
</extension>

<extension
    name="Text Importer"
    description="Collaborate on text files in wave!"
    thumbnailUrl="http://media-api.appspot.com/.../yellow-sign.png">
  <author name="Google Wave Media Team"/>
  <menuHook
    location="TOOLBAR" text="Import from Text File"
    iconUrl="http://media-api.appspot.com/.../yellow-icon.png">
    <addParticipantsAction>
      <participant id="media-api+import-txt@appspot.com"/>
    </addParticipantsAction>
  </menuHook>
</extension>
```

Summary of Changes

Summary of API Changes

- Robots
  - Attachment element
    - mimeType, attachmentUrl, data
- Gadgets
  - attatchment_url state injection
- Extensions
  - import
  - export
  - viewer

Google I/O 10

Q & A

To find out more visit:

- http://code.google.com/apis/wave/

Thanks!



Google 10 I/O

View live notes and ask questions about this session on Google Wave

http://bit.ly/wave-media-api


Google™
10