

# Offline in HTML5

Petr Kunc, LaSArIS 18. 9. 2013

# WHY?

- offline **web** application
- migration from desktop
- cooperation
- mobile connection
- “bad click”
- syncing?

# HOW?

- WebStorage
- Databases
- FileAPI
- Offline Detection
- Application Cache

# Web Storage

- store dynamic data
- desktop apps
- web apps on server
- Cookies, userData, LSO, Gears, Air, Flash,  
...
- localStorage and sessionStorage

# Interface

```
interface Storage {  
  readonly attribute unsigned long length;  
  DOMString? key(unsigned long index);  
  getter DOMString getItem(DOMString key);  
  setter creator void setItem(DOMString key,  
  DOMString value);  
  deleter void removeItem(DOMString key);  
  void clear();  
};
```

# Example

```
if (window.localStorage) {  
  //these three examples of setting data is  
  equivalent  
  localStorage["key"] = "data";  
  localStorage.setItem("key", "data");  
  localStorage.key = "data";  
  //these three examples of getting data is  
  equivalent  
  output = localStorage["key"];  
  output = localStorage.getItem("key");  
  output = localStorage.key;  
}
```

# Databases

- Dead WebSQL
- IndexedDB
- object store mechanism
- cursors, ranged queries, indexes
- asynchronous API

```
//standard synchronous data access
//usually developers obtain data by assigning
//a returned value of some method to a
//variable
result = database.get("key");

//Indexed DB asynchronous request:
var request = objectStore.get("key");
request.onerror = function(event) {
//when IndexedDB fails when obtaining the key,
//it triggers the onerror function
};
request.onsuccess = function(event) {
console.log("Value of key is " + request.
result.value);
};
```



# File API

- sandbox
- dialog window
- asynchronous

`readAsBinaryString`

`readAsText`

`readAsDataURL`

`readAsArrayBuffer`

# Online/offline

- `window.navigator.onLine`  
(2 events - online and offline)

- appcache
- Comet, polling
- WebSockets

# Application cache

- main offline technology
- caching in browsers
- explicit control
- list of files and rules
- linked in html tag

```
<!DOCTYPE HTML>
```

```
<html manifest="manifest.appcache">
```

```
</html>
```

# Browsers

Browser	Since version
Internet Explorer	10
Firefox	3.5
Chrome	4
Safari	4
Opera	10.6

# Limits

- Limits vary in browsers (from 10 MB to unlimited)
- Reactions vary in browsers (from prompt to decline)

# Structure

- First line

CACHE MANIFEST

- blank lines
- comments (#)
- section headers
- section data

CACHE MANIFEST

#VERSION: 2013-04-18 16:45

#the explicit section is default, no

#need for CACHE:

CACHE:

styles/default.css

scripts/main.js

<http://code.jquery.com/1.9.1.min.js>

FALLBACK:

/images/avatars/ images/offline.png

/ /offline.html

NETWORK:

\*

SETTINGS:

prefer-online

NETWORK:

# login page should not be cached

# credentials need to be sent to server

login.php

# forbid to cache any resources in authorized  
section of web

/authorized

# access to any API URL has to be online

<http://api.twitter.com>

- ASTERISK!!!



FALLBACK:

#everything from /images/avatars

#prefix link to default image

/images/avatars/

/images/avatars/default.png

#every other resource from /images/

#prefix link to offline.png

/images/ /images/offline.png

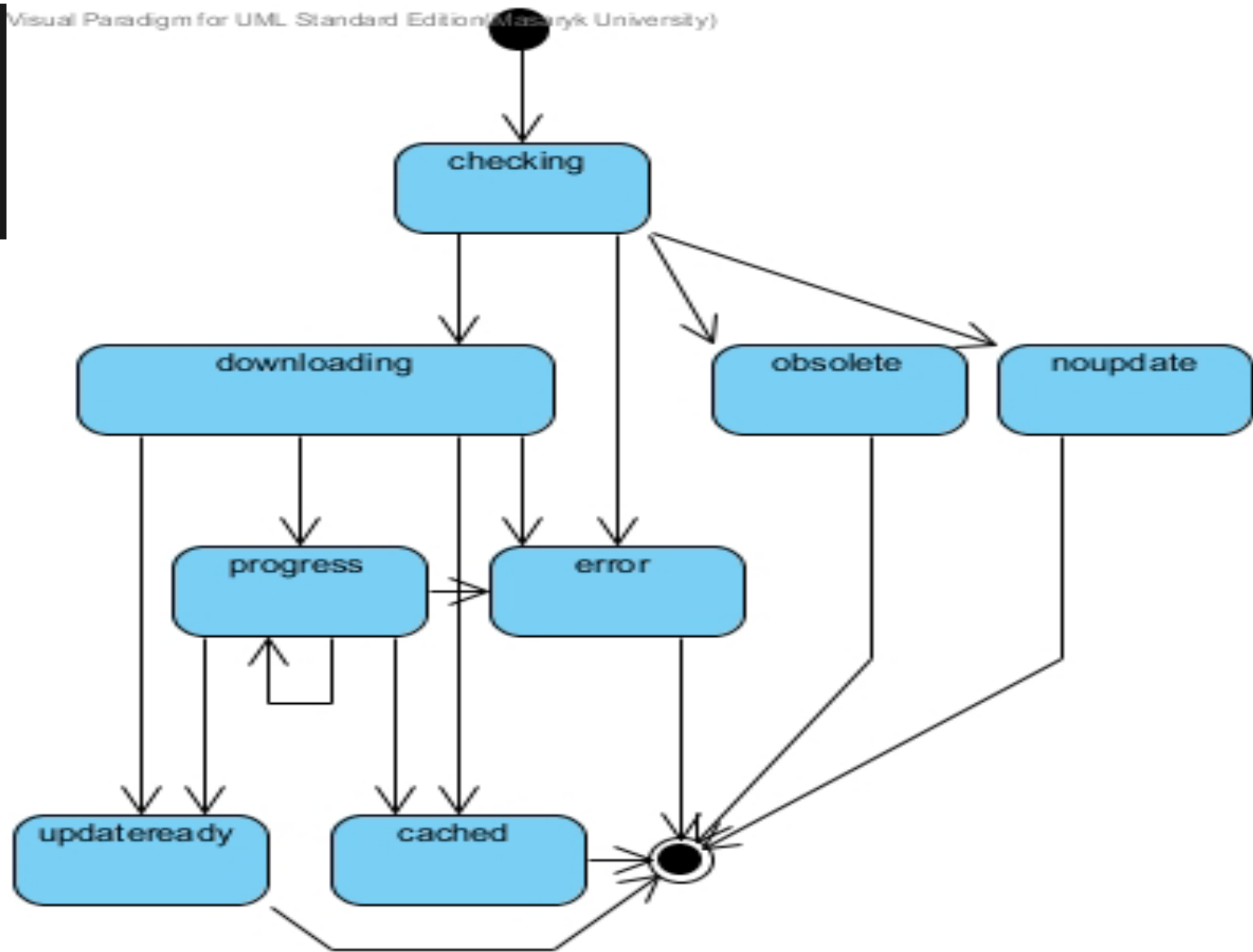
#any URL not matching the two above

#link to offline.html

/ /offline.html

# Settings

- Currently only one  
PREFER-ONLINE



# Advantages

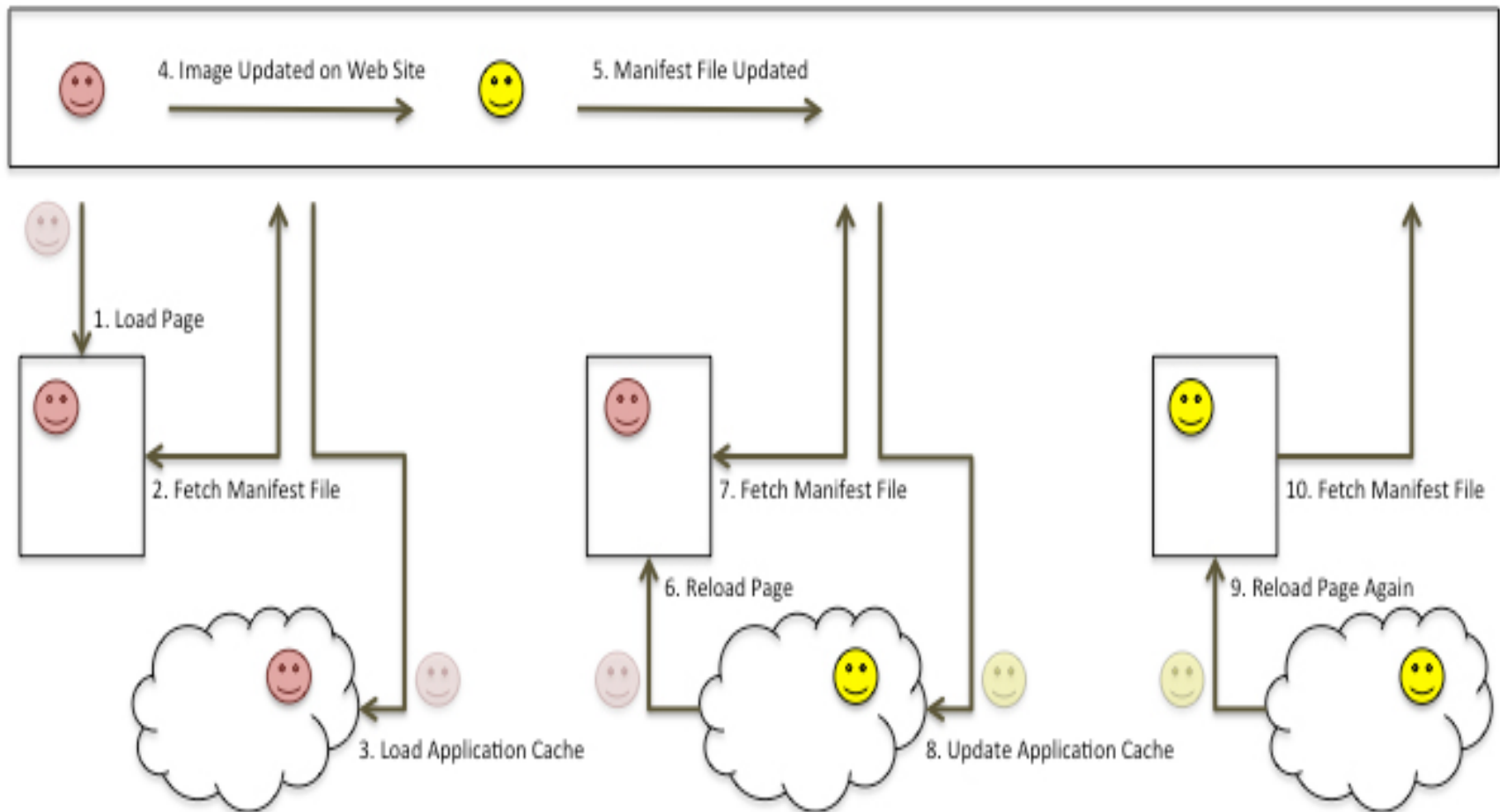
- Offline
- Fast
- Saves bandwidth

And Disadvantages:

# Change of Resources

- just when the manifest changes
- 304 Not Modified
- version comments
  
- never far-future cache these files

# Double Reload



# Modularity

- one file for application
- error-prone lists
- process controlling
  
- basic static
- or modern JS MVC apps

# Proposed solution

- LESS CSS inspired
- extension to original language with processor
  
- imports
- filters (regexp, glob)
- extensible
- language check
- file check
- automatic version controls



# Thanks!

Your questions?