Introducing… WBTS
[Web Browser Testing System]

BlackHat 2011

August 3rd, 2011

VERACODE

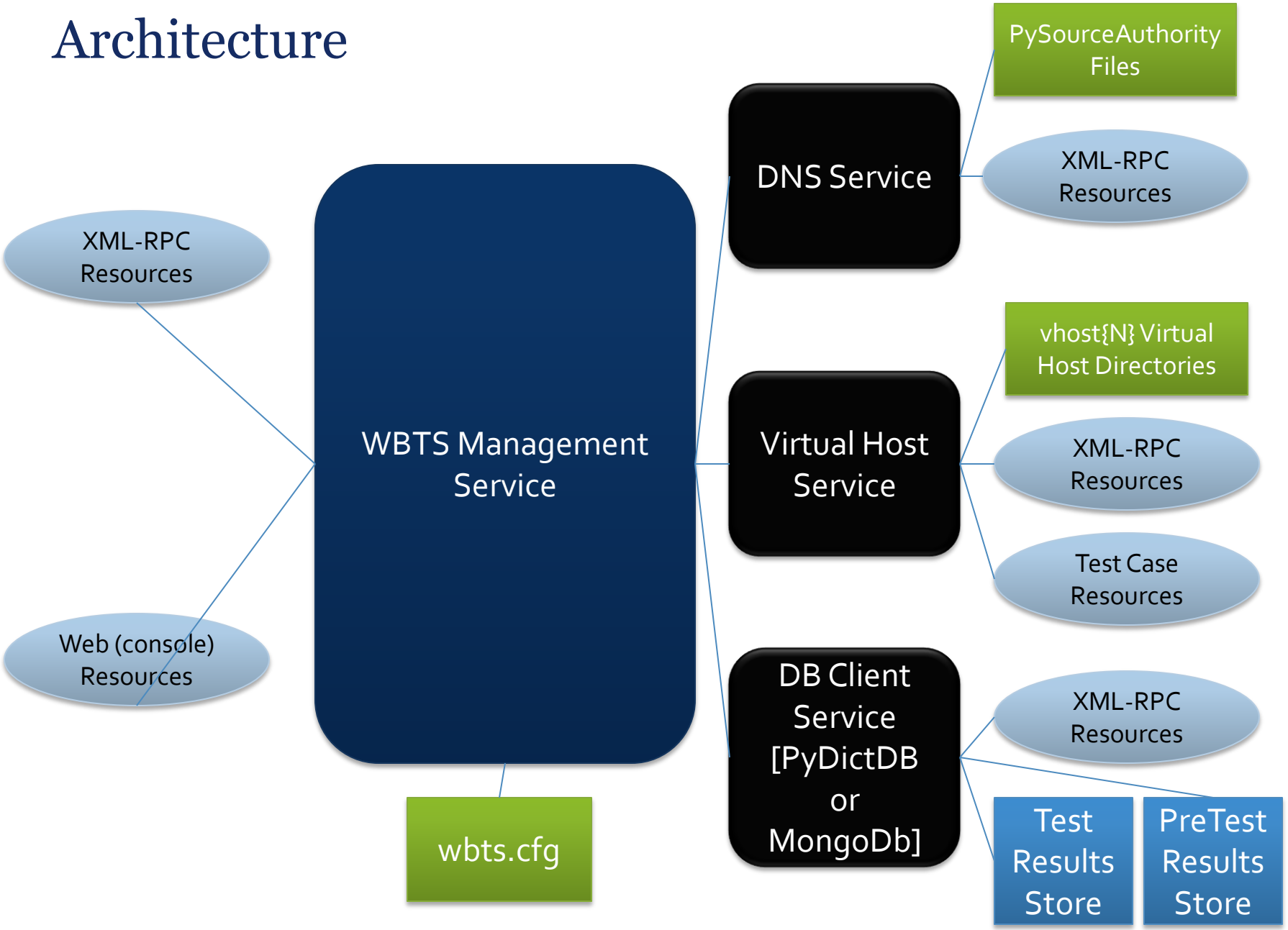
Uh, what is WBTS?

- Ties in multiple services: Virtual Hosts (HTTP/HTTPS), DNS, remote logging and management all built into a single system.
- Comes with the "blade" automation system because running tests manually, kind of sucks.
  - Blade is just the name for the in-browser (JavaScript) and external (Python) scripts.
- Manageable from a web interface (or vim… or emacs if you're old school^W).
- All services were built using Python's stupid fast asynchronous framework, Twisted. (http://www.twistedmatrix.com)
- *Sort of* supports 3rd party processors (php/perl)
  - I say sort of because I don't really like either and haven't tested it beyond the fact that they run simple test code.

Where can I get it?

- Get the code!
  - http://code.google.com/p/wbts/
  - http://code.google.com/p/wbts-runner/

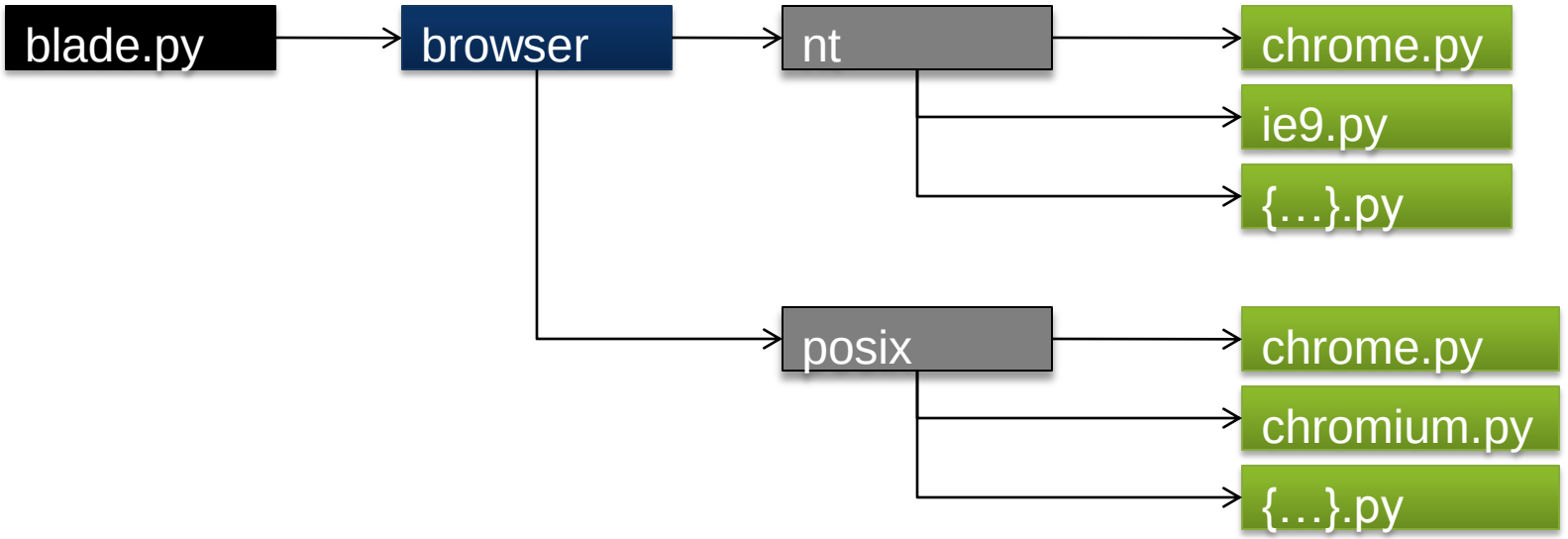
Architecture



The blade automation system (quick overview)

- The blade system is for automating browsers to run through our manually created test cases.
- Consists of server side resources and client side scripts and tools.
- Server side code is simply an XML-RPC interface to list test cases.
- Allows you to exclude individual files or sub directories.
- Implemented in two ways: JsBlade (in browser) and blade.py.
- JsBlade runs through tests by updating an <iframe> src attribute.
- blade.py loads the browser and uses UI automation routines.
  - pywin32 for Microsoft Windows
  - LDTP for Linux

blade.py structure



blade.py current browser support

Browser	OS	Method
Chrome	Windows	Pywin32
IE9	Windows	Pywin32
Chrome	Linux	LDTP
Chromium	Linux	LDTP

The blade runner process

- Generate Unique ID
- Append it to test case id (the url/filename)
- Load the test case (test runs at this point)
- Call getTestProgress
- No results?
- Poll every few seconds, 4 times.
- If getTestProgress never returns results, call failTest
- Go to next case…

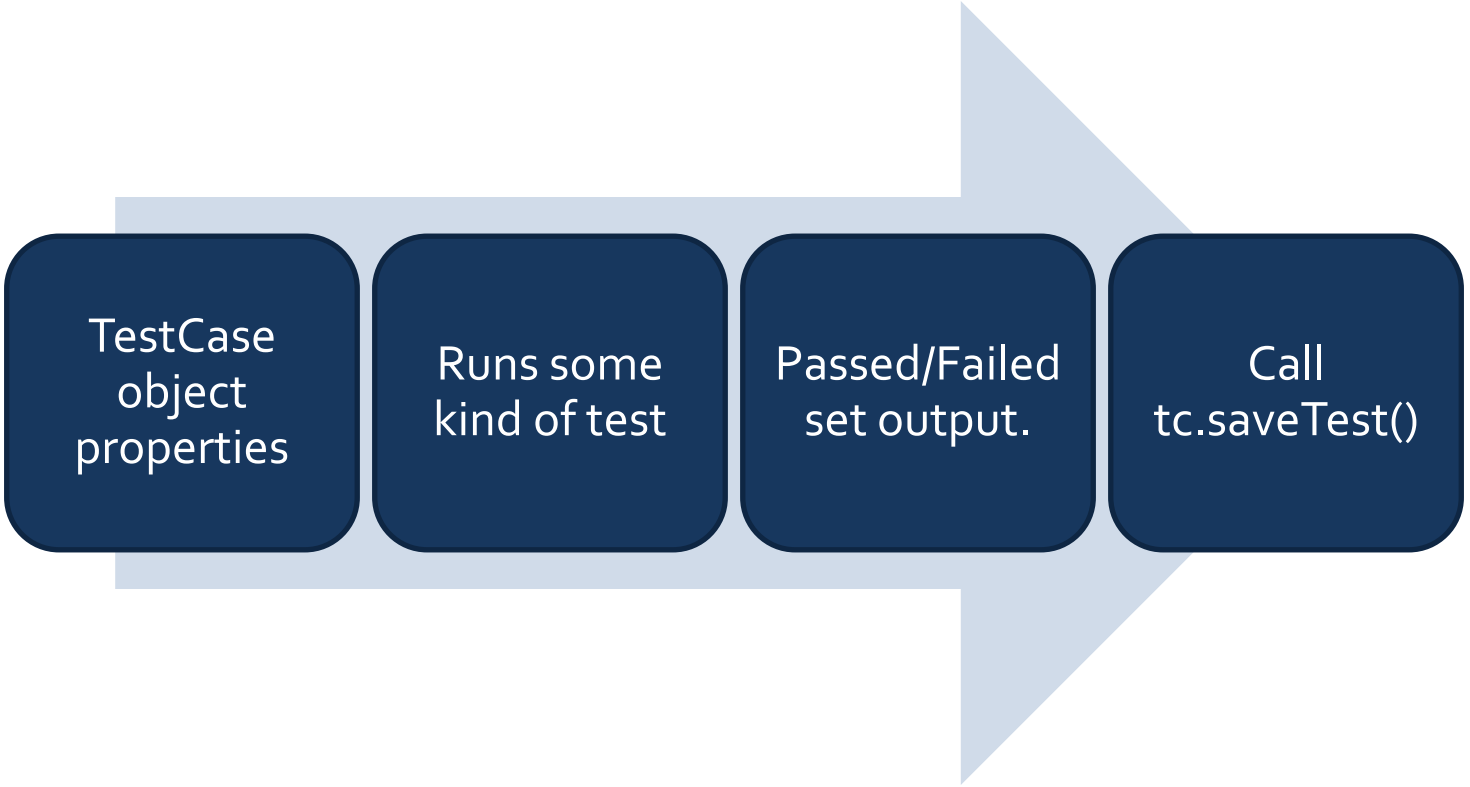
WBTS Test Cases

- So, what does a WBTS test case look like?
- A test case is just a JavaScript object with various properties.
- Two types: simple and pretest.

Property	Description
input	Either the exact test case input, or a rough outline of what it is.
description	A basic description of the purpose of the test.
expected result	What the expected result of the test will be
result	The result of the test
output	The exact output of the test, like the caught exception and it's message
test passed	A true or false value determining if the test passed or not.

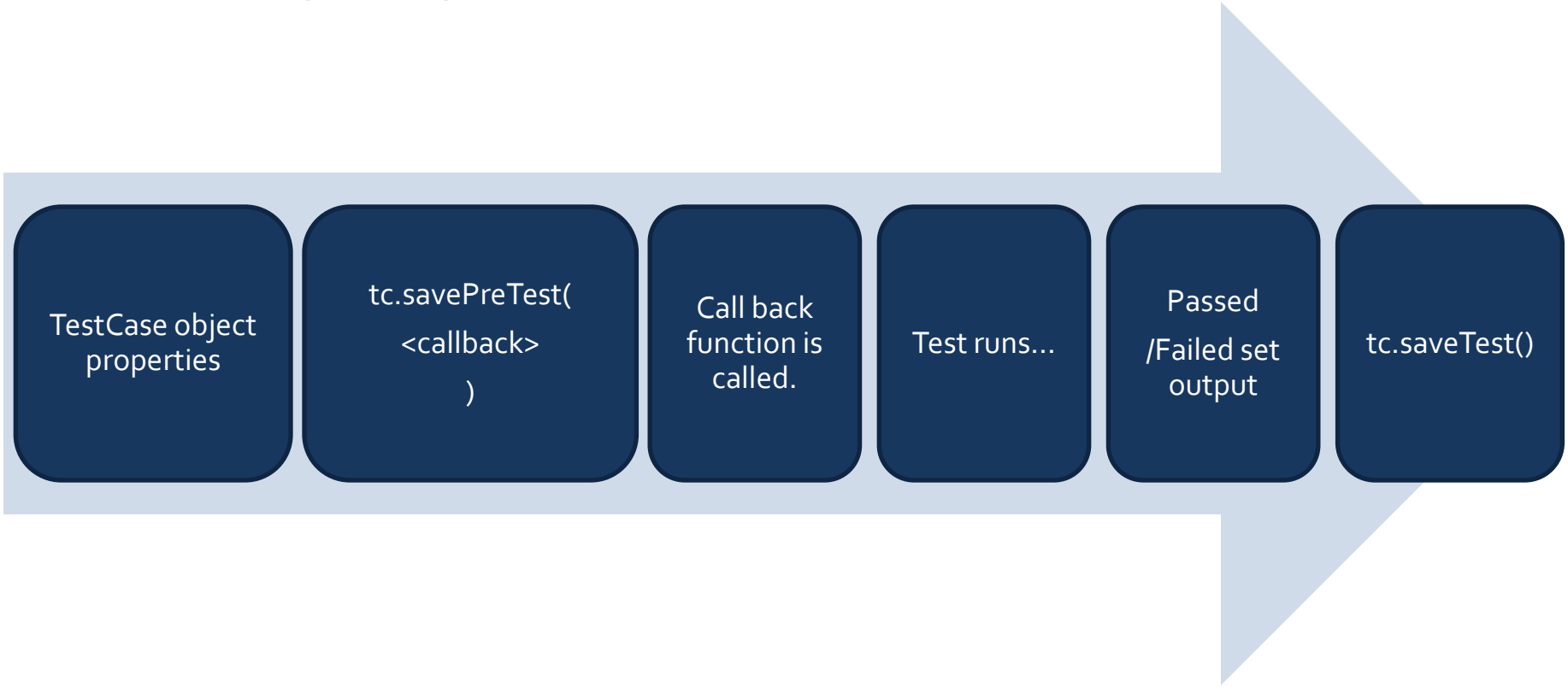
Test Cases – (continued)

- A "simple" test case:
- See http://wbtshost/shared/templates/simple-testcase.html
- Used for the majority of test cases, where the chance of the interpreter bombing, or something failing horribly is low.



Test Cases – (continued)

- A "pretest" test case:
- See http://wbtshost/shared/templates/pretest-testcase.html
- Used only when the chance of the interpreter bombing, or something failing horribly, will most likely occur.



Using WBTS to help you find bugs.

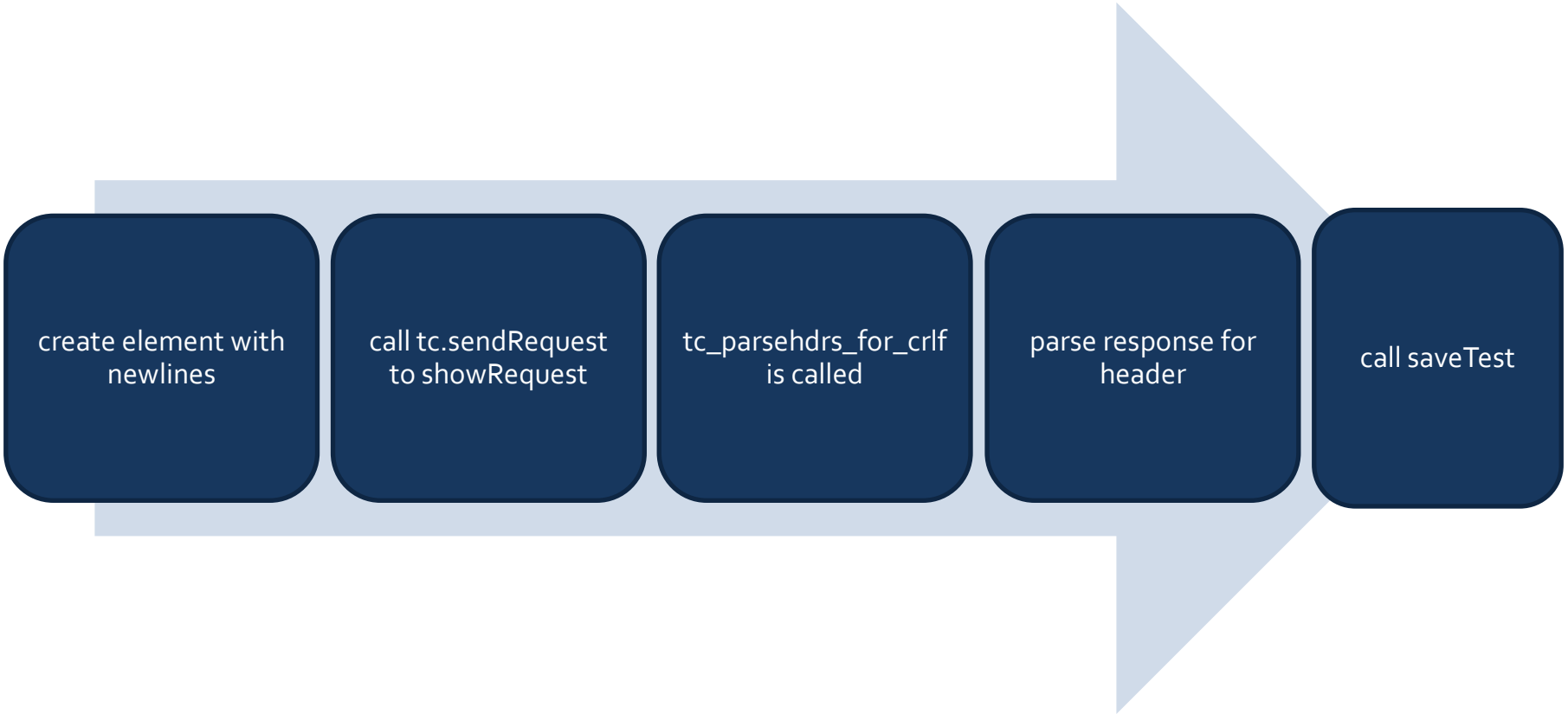
- Besides memory corruption, what else is there?
  - Same Origin Policy Bypass – attacker.com meet bank.com
  - Header Injection – X-MOBILEID=NOTME123
  - Cookie handling issues – Set-Cookie: alljpcookie=1234; domain=.co.jp;
  - Parser Issues - <scr\xef\xbb\xbfipt>alert(123)</script>
  - Architectural/Design Issues – CSS knows where I've been :/
  - File Handling Vulnerabilities – input type is a radio, no wait… now it's a file! oh noes! document.forms[0].submit(); *
  - URI Handler Issues –mailto://"%20--run="something.exe"
  - Protocol Implementation Issues – My cert comes with null bytes
  - Trust Issues with Component Interactions – Applets are Safe, Really

SOP Bypass Testing with WBTS

- Make use of the HeaderSetter resource to create arbitrary response headers.
  - Pass the header data as a query parameter to any resource/testcase:
    - http://attacker.com/shared/resources/somefile.html?encode=b64&headers=aGVhZGVyMTogdmFsdWUxCmhlYWRlcjI6IHZhbHVlMg==
- Use the RedirectResource to create custom redirects.
  - Supports any response code, not just the default 301.
  - Can include funky bytes by base64 (encode=b64) or url (encode=url) encoding:
    - http://attacker.com/redirect?code=302&loc=http://victim.com/forbidden.html
- Use the TestCase objects methods for testing cross domain accessibility:
  - tc.readOriginData(iframe, origin)

Header Injection with WBTS

- Use WBTS's /showRequest resource to view raw request data.
- Make use of the TestCase helper functions to parse/analyze such as:
  - tc_parsehdrs_for_crlf(xhr, tc);



Testing Cookie Handling with WBTS

- Use the showRequest resource to view the raw request data.
- Use sub-domains and the WBTS DNS service for testing domain/scoping related issues.
- The TestCase object will automatically clear cookies before and after tests begin.

Testing Parser Issues with WBTS

- Make use of the getOutput resource as it allows insertion of arbitrary text to be echoed back.
  - Example: http://attacker.com/getOutput?q=asdf
- Use savePreTest if you think the parser will break prior to your test executing.
  - Look at /testcases/misc/parser/utf-8-bom-in-script-tag.html as an example.

File Handling Vulnerabilities

- Types of file handling functionality to test.
  - HTML Forms
  - File Download
  - Cookies (Yes Cookies)
  - Caching
  - file:// URI handler
- Newer APIs that deal with files should definitely be targeted.
  - File API
  - File API: Writer
  - Indexed DB
  - WebStorage
  - Web SQL

Testing File Handling with WBTS

- Unfortunately…. Unable to handle file uploading easily via automation.
- Use the formSubmit (test_type =“upload”) resource for viewing the uploaded files contents.
- Use the ability to set arbitrary headers (HeaderSetter) for testing file downloads:
  - http://attacker.com/shared/resources/4x4.png?headers=content-disposition:%20attachment
  - * Use encode=b64 if you need to pass attachment; filename=…

Testing URI Handlers with WBTS

- Requires rather manual processes.
- Use the RedirectResource or HeaderSetter resources to create redirects and custom headers to test URI handlers.

Protocol Implementation Issues with WBTS

- WBTS won't really help with testing protocols as it is at a much lower level.
- WBTSHTTPChannel ***could*** be extended/modified for testing low level
- Plans to implement…
  - An FTP server
  - A WebSockets service

Testing Trust Issues in Browsers with WBTS

- Use the DNS system for testing cross origin leakiness.
- Use the various resources to test how components react to different headers.