Comparing JSON in RFC 4627 to ECMA-262
IETF 86, Orlando

Paul Hoffman
Parts of RFC 4627

- Grammar
  - “JSON text” as a noun
- Parser
- Packaging
  - Encoding
  - MIME media type
- These three parts are not completely segmented in the RFC
Parts of ECMA-262, version 5.1

• Mostly about ECMAScript/JavaScript, but part is about JSON
• 15.12: “The JSON Object”
  – “JSON text” as a noun
• 15.12.1: JSON Grammar
• 15.12.2: parse function
• 15.12.3: stringify function
Comparing the grammars

- RFC 4627 limits a JSON text to being either an object or an array; ECMA-262 doesn’t have this limit
- Both specs allow objects to have duplicate names within them; in RFC 4627, there is a SHOULD against this
Comparing the parsers

• In RFC 4627, it is unclear how an object that has two names the same should be parsed; in ECMA-262, it is specified as the last one wins
• RFC 4627 allows a parser to accept non-JSON forms or extensions
• ECMA-262 parser has an additional function for filtering
Comparing the packaging

• A JSON text in ECMA-262 is in a variable; in RFC 4627 it is in a MIME object
• In both, a JSON text is a string of bytes which encodes Unicode characters
• The text encoding in RFC 4627 is probably part of the packaging
Other bits of difference

- RFC 4627 refers to version 3 of ECMAscript
- ECMA-262 has a stringify() function
- The security considerations in RFC 4627 has some JavaScript-specific regex for passing to eval()