**Topic:**

Node.js – Novice to Ninja

**Contact Information:**

**Name: Guillaume Brossard, David Richer**

**Location: Montreal**

**Division & Team:** **M&E – Web Services – Production Management**

**E-mail:**  **guillaume.brossard@autodesk.com; david.richer@autodesk.com**

**Speaker/Exhibitor Biography:**

|  |
| --- |
| Guillaume Brossard previously worked at GRIP Entertainment, where he designed and built an artificial intelligence behavior tree system for the video game industry, and at Bluestreak Technology, developing on a real-time Adobe Flash compatible rendering engine for embedded devices.  David Richer also previously worked at GRIP Entertainment where he designed and built a real-time character system for video game digital extras. He has over 10 years’ experience in C++.  Both speakers are experienced C++ programmers that became JavaScript enthusiasts. They are working together in the M&E Asset-Based Collaboration team, creating a tool bringing to artists and production managers the power of the cloud. |

**Proposal Abstract:**

As part of the Autodesk business transformation to the cloud, the Web Services group within the M&E division has the mandate to smooth the transition to the cloud for various product groups. Some key goals within this initiative include:   
- Ensuring the shortest learning curve possible for native developers making transition to the cloud.   
- Allowing components to be shared among teams.   
- Facilitating fast development cycle of micro-services.  
  
Node.js has been identified as a technology with a broad applicability to the challenges faced by the Web Services group. This presentation is for developers by developers, and will demystify, using real-world examples, what is Node.js and why it should be considered over alternative technologies. Attendees will learn how to take advantage of a full JavaScript stack composed of GitHub, Node.js, npm, Grunt.js and NoSQL databases to develop sharable components at a ninja-fast pace.

**Intended Audience & Prerequisites:**

* Web services and web application developers accustomed to non-Node.js technologies.
* Web developers that want to learn how to leverage Node.js packager (npm) for content sharing.
* Developers new to web services.
* Anyone wishing to learn about Node.js and why it is gaining traction in the market.

**Presentation outline:**

General track: Don’t fight the current, follow the flow

|  |
| --- |
| * M&E Cloud Services Context (1-2 min) [GB] * Node.js Overview (2-4 min) [GB]   + - 1. Evented, single threaded I/O build over V8 engine (Chrome)       2. JavaScript       3. Module based       4. Native C++ modules * Why Node.js (4-5 min) [GB]   + - 1. Same language client-side/server-side       2. Object model consistency (JSON)       3. Asynchronous by nature       4. Large collection of quality open source modules       5. Express: Minimal web application framework       6. Node.js Express Hello World * Node.js for Enterprise (1 min) [DR]   + - 1. Developer environment       2. Sharing Components between teams       3. Deployment          1. Local          2. VM          3. Amazon * npm: Node.js Package Manager (2-3 min) [DR]   + - 1. Node.js modules (JS and native)       2. Configuration (package.json) * Developer Environment (2-3 min) [GB]   + - 1. Leveraging npm for our needs          1. Npm script          2. One step installation          3. linkDependencies, webDependencies       2. Grunt.js – Javascript Task Runner for Node.js          1. Tests, Static Analysis, Code Coverage          2. Continuous Build Integration * Sharing node modules between teams (2-3 min) [GB]   + - 1. Component-based architecture       2. Sharing Components without npm registry          1. Npm + Git * Deployment (5-7 min) [DR]   + - 1. VM          1. Vagrant          2. Chef       2. Amazon          1. Cloud Formation          2. Chef       3. Deployment Set-up (VM + Amazon)          1. Gitolite / Chef / Cloud Formation / Vagrant          2. Configuration Component (nconf) * Conclusion (1-2 min) [DR]   1. Full JavaScript stack use-case: Web Services and Production Management      + 1. Versioning System: GitHub        2. Package Manager: npm        3. Automation Tool: Grunt.js        4. Database: MongoDB/CouchDB        5. IDE: WebStorm        6. Release Tool: npm/Git        7. Deployment: Chef/Git   2. Hipster technology gaining credibility      + 1. Node modules           1. Databases: MongoDB, CouchDB, PostgreSQL, etc.           2. Amazon : aws-sdk, awssum           3. Build automation : Grunt.js        2. Chef recipes (deployment)        3. WebStorm – Node.js ready * Additional material   + - 1. Code organization       2. Release Management       3. ~~Pros and~~ Cons       4. QA (JSLint, Code Coverage, Jasmine, Chai, Karma, etc.)       5. Librairies (underscore, express, connect, request, async)       6. Technology Shopping List |

**Additional materials:**

|  |
| --- |
|  |