

Integrating .NET Code with AutoCAD I/O to Add Design Intelligence to Your Website

Kean Walmsley

Software Architect, Autodesk

[@keanw](https://twitter.com/keanw)

Class summary

Over the last decade or so, software developers have amassed a significant amount of intellectual property harnessing AutoCAD's .NET API. AutoCAD I/O enables standard AutoCAD software commands, as well as those implemented in .NET, to be executed in the cloud, generating results that you can integrate into your own business-to-business or business-to-customer website.

This class will take a concrete example of a .NET application creating custom jigsaw puzzles inside AutoCAD software. During the class we will show how to move the core implementation to AutoCAD I/O via the Core Console, and then make use of this to power a new business-to-customer website. Potential customers will be able to specify custom designs for jigsaw puzzles and visualize the results before finalizing their orders.

Key learning objectives

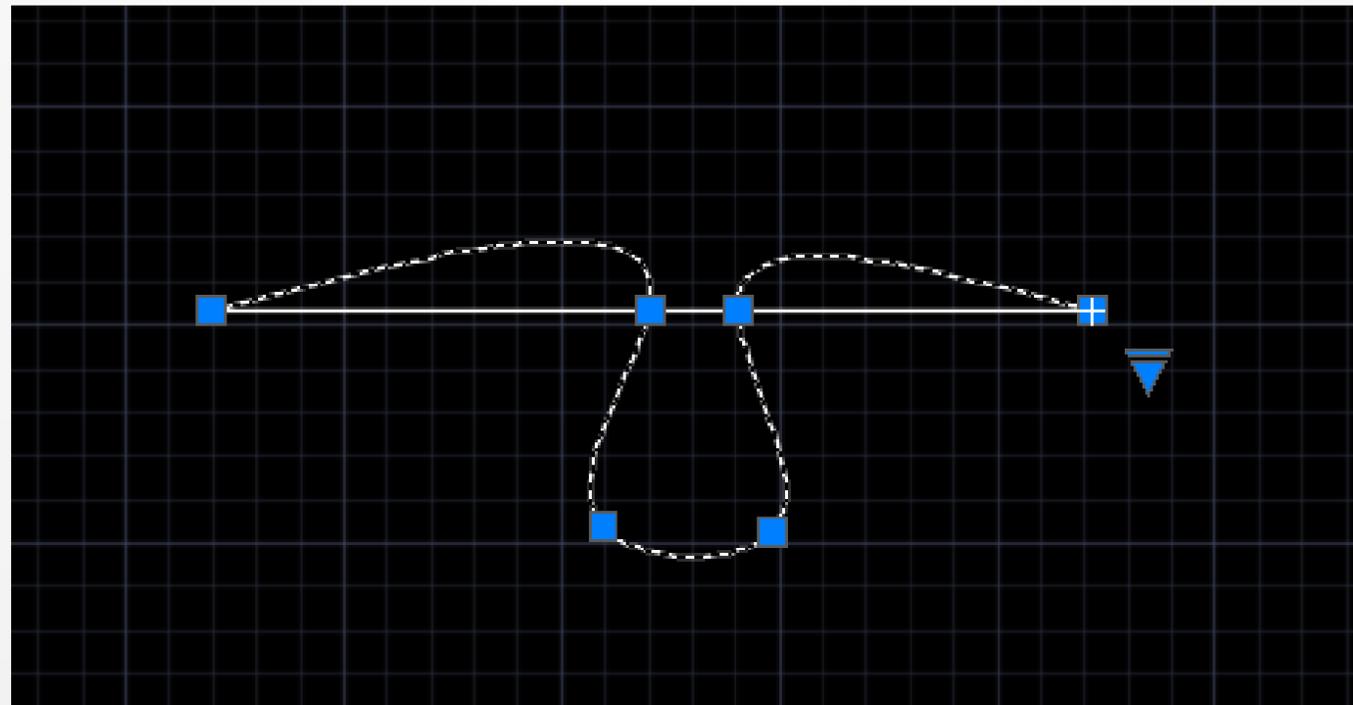
At the end of this class, you will be able to:

- Get started with the AutoCAD I/O service
- Learn how to take existing AutoCAD .NET applications and create "core" modules
- Learn how to drive AutoCAD I/O with custom .NET application from a website or web service
- Learn how to integrate the results from AutoCAD I/O, displaying them via the web

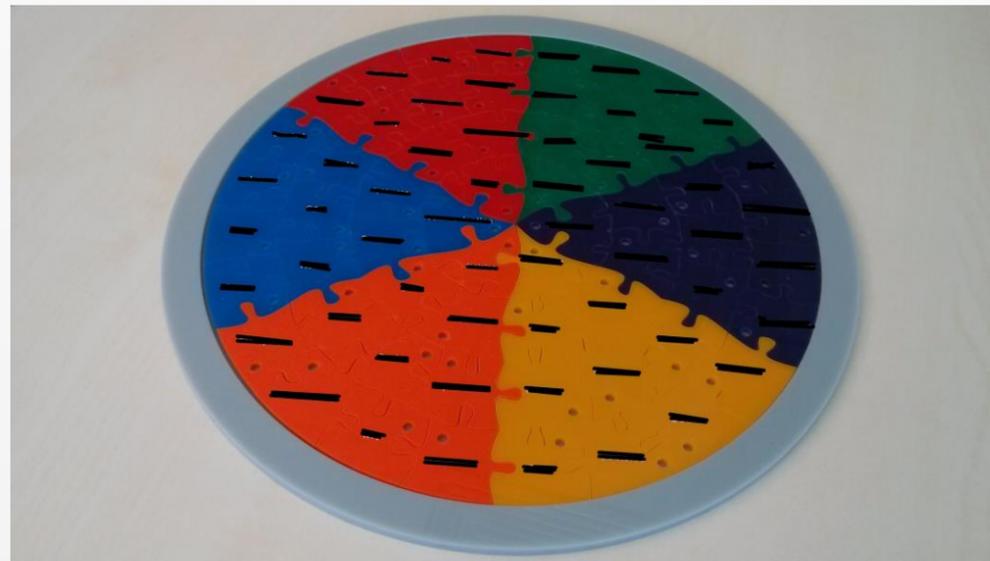
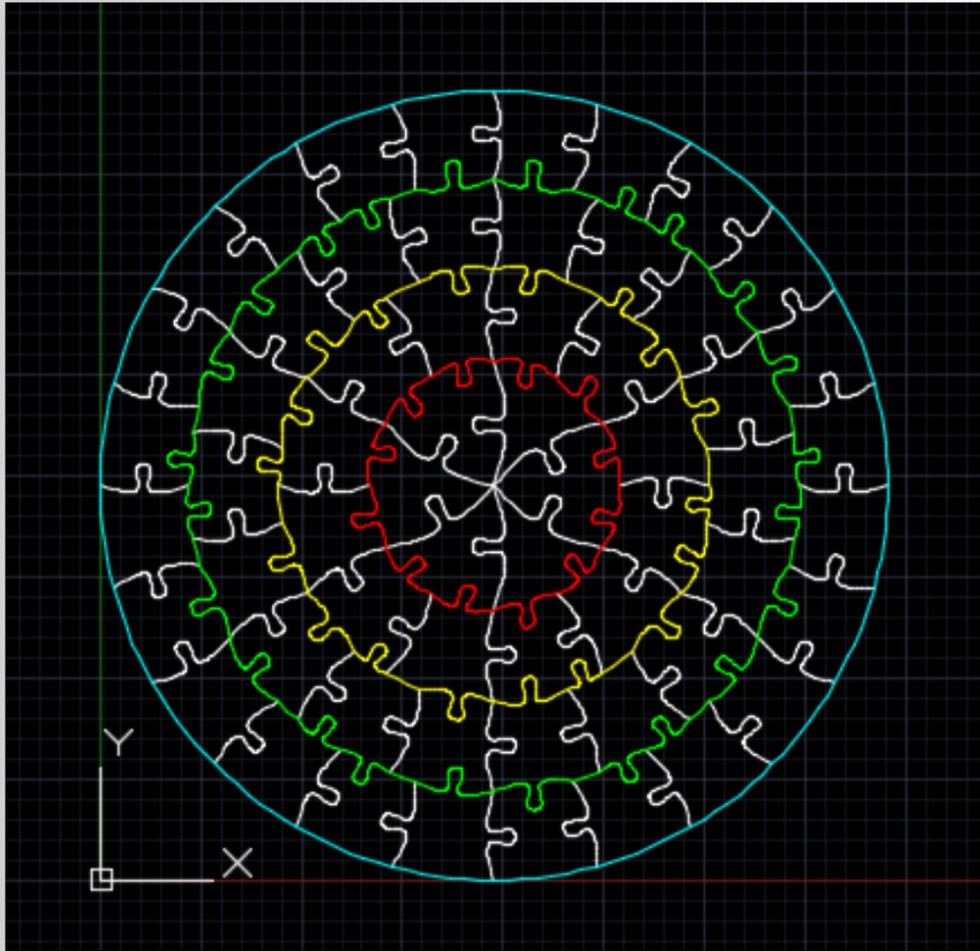
A puzzling problem

Inspiration

- Our scenario was inspired by an internal project
 - A colleague was looking for a 60-piece, circular jigsaw puzzle
 - After searching the web in vain...
 - ... I wrote some code to convert curves to jigsaw patterns



The project



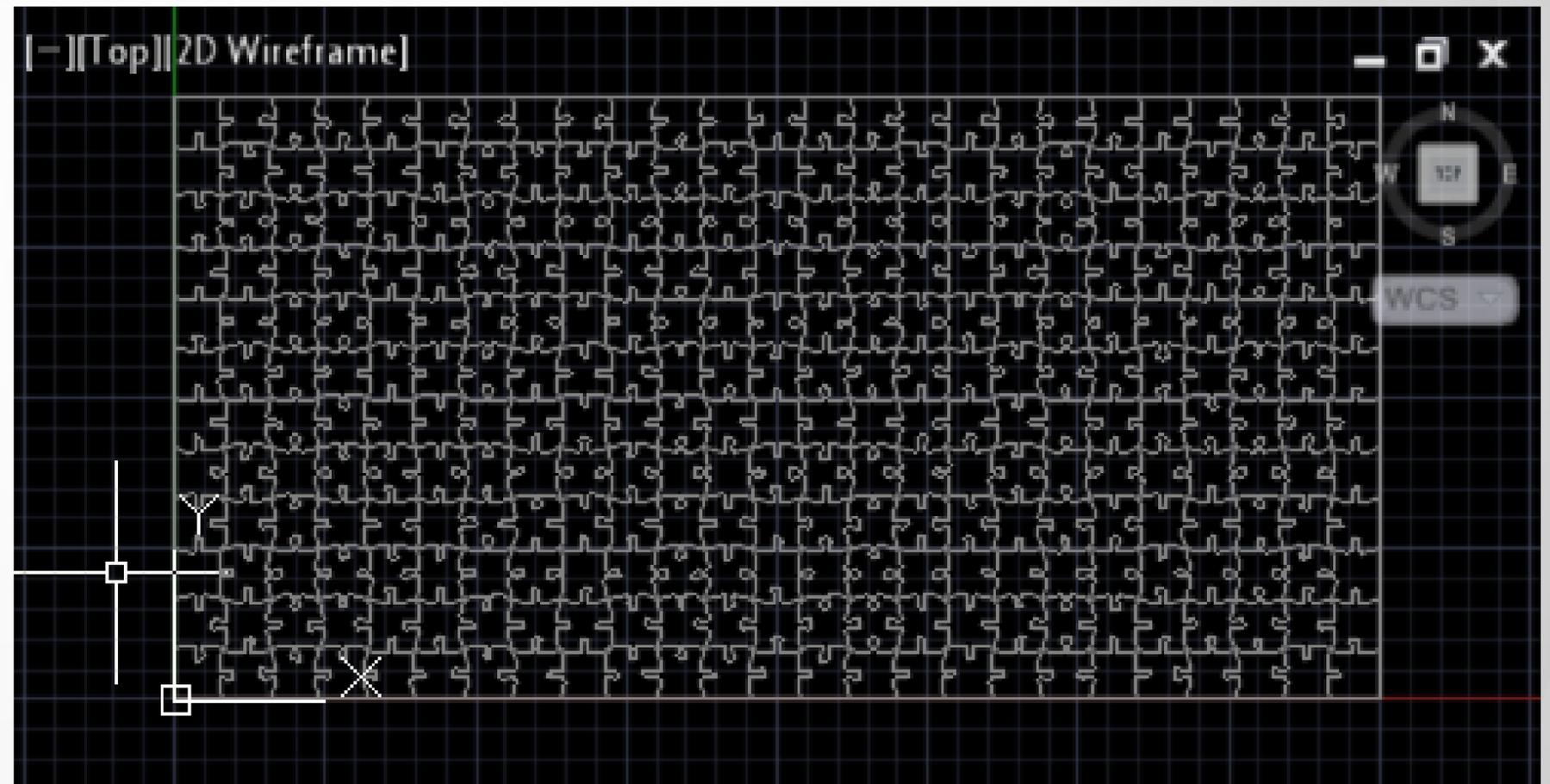
A command to generate a rectangular puzzle

- The JIGG command takes....

- Width
- Height
- Number of pieces

- And generates....

- A random cut pattern



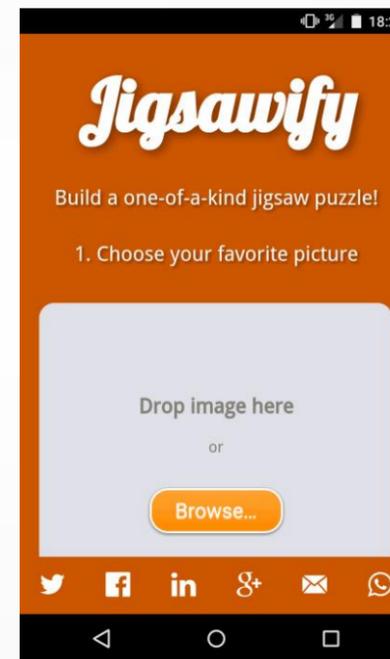
Taking it to the next level

- Engraving with a laser cutter is slooooooow
 - There's also a manual step between phases
 - Engraving is directly from a bitmap, not from DWG/DXF data
- The approach
 - Replace simple engraving by extracting edges from the bitmap
 - Add these as 2D Solids into the DWG/DXF
 - It turned out this didn't work with our laser cutter
 - Diagonal lines across the square worked well
 - Can OVERKILL them to make them contiguous

And so Jigsawify was born

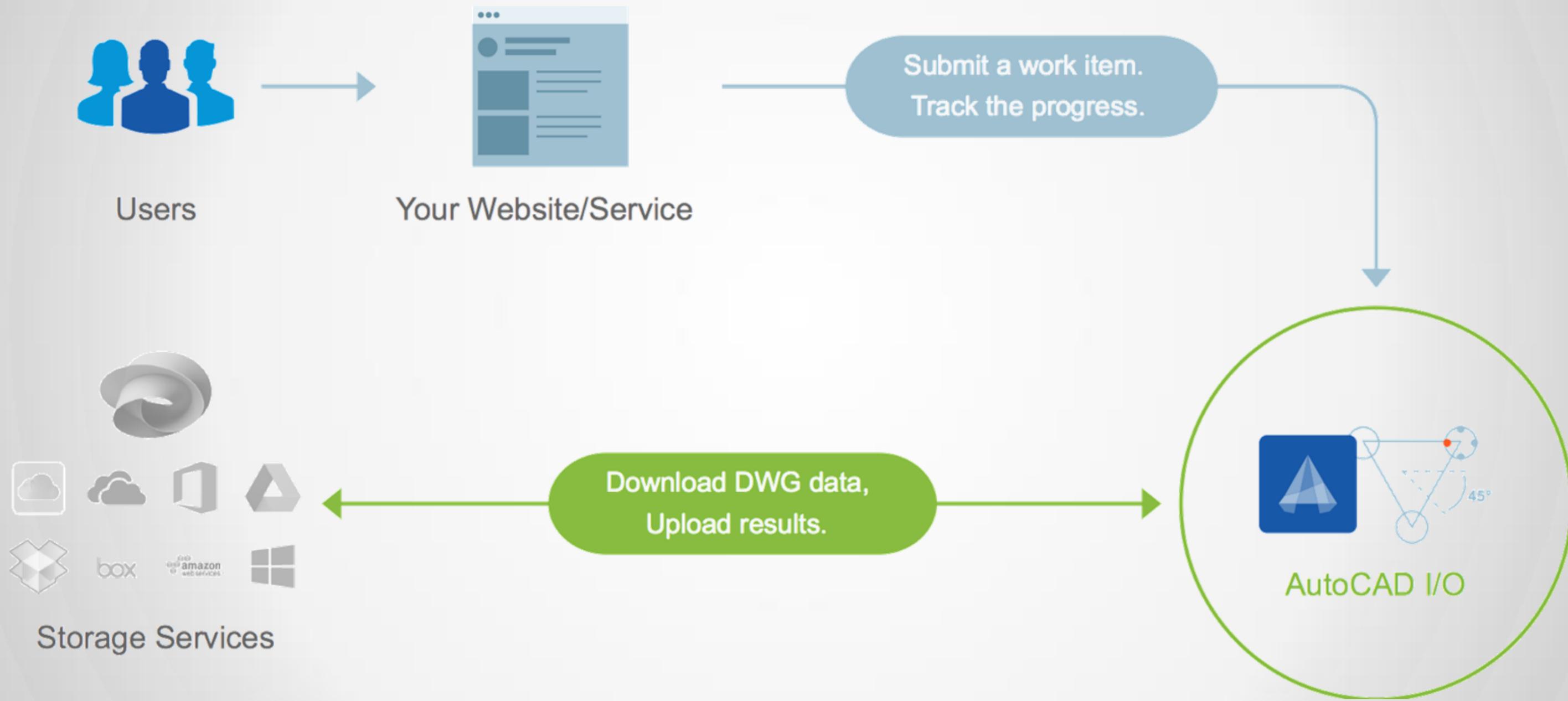
■ Jigsawify.com

- Jigsaw puzzles from photos
 - Adjustable edge detection
- DWG & DXF output
 - Ready to drive a laser cutter



Getting started with AutoCAD I/O

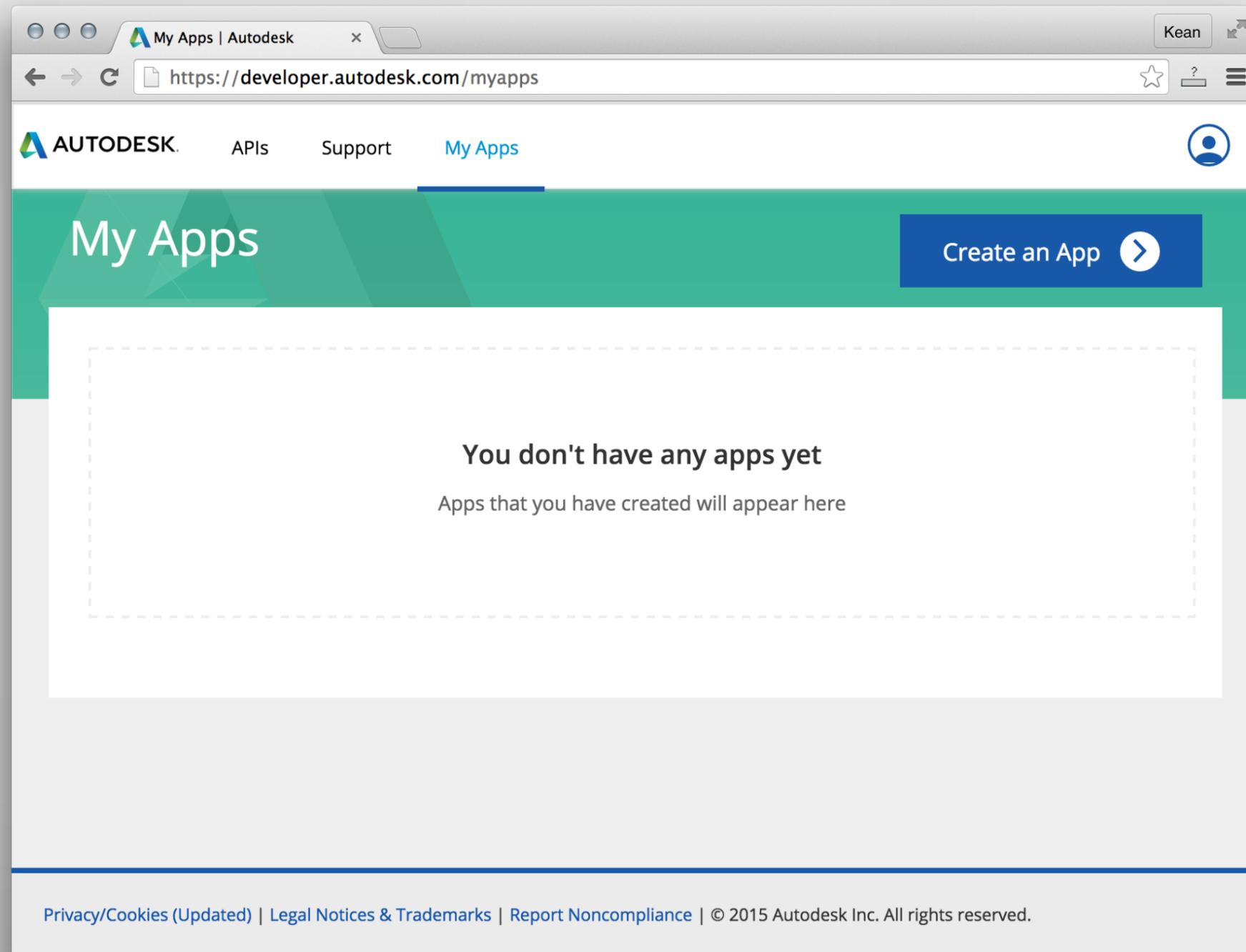
What is AutoCAD I/O?



OK, but what is it, really?

- A web service...
- Allowing you to batch process DWGs in the cloud...
- Running standard or custom AutoCAD commands...
- Inside the Core Console, but with additional security...
- Generating DWG, DXF or other output formats.

Getting started: credentials!



Creating an app

The screenshot shows a web browser window at the URL `https://developer.autodesk.com/myapps/AU%20IO%20Test/view`. The page title is "AU IO Test" and a green notification box says "Your new app is successfully created!". Under the "APIs" section, there is one API listed: "AutoCAD® I/O API". Below this, the "About this app" section shows it was created on 12 Nov 2015. The "Consumer Key" is `uIY0eiupH6764WU7tzNWXuTHIpDA55AJ` and the "Consumer Secret" is masked with asterisks, with "Show" and "Regenerate" links. The "App Name" field is also visible.

Never share these!

**Includes embedding
directly in a DLL**

**Hide them behind
a web-service**

Start small with a simple sample

- AutoCAD I/O samples
 - github.com/Developer-Autodesk/AutoCAD.io
- Simplest sample to get started
 - github.com/Developer-Autodesk/autocad.io-simplest-Csharp
- Clone it locally and insert your credentials
 - Running it will create a WorkItem for an existing Activity
 - PlotToPDF: generates a PDF file for a given DWG

Creating core modules for your .NET apps

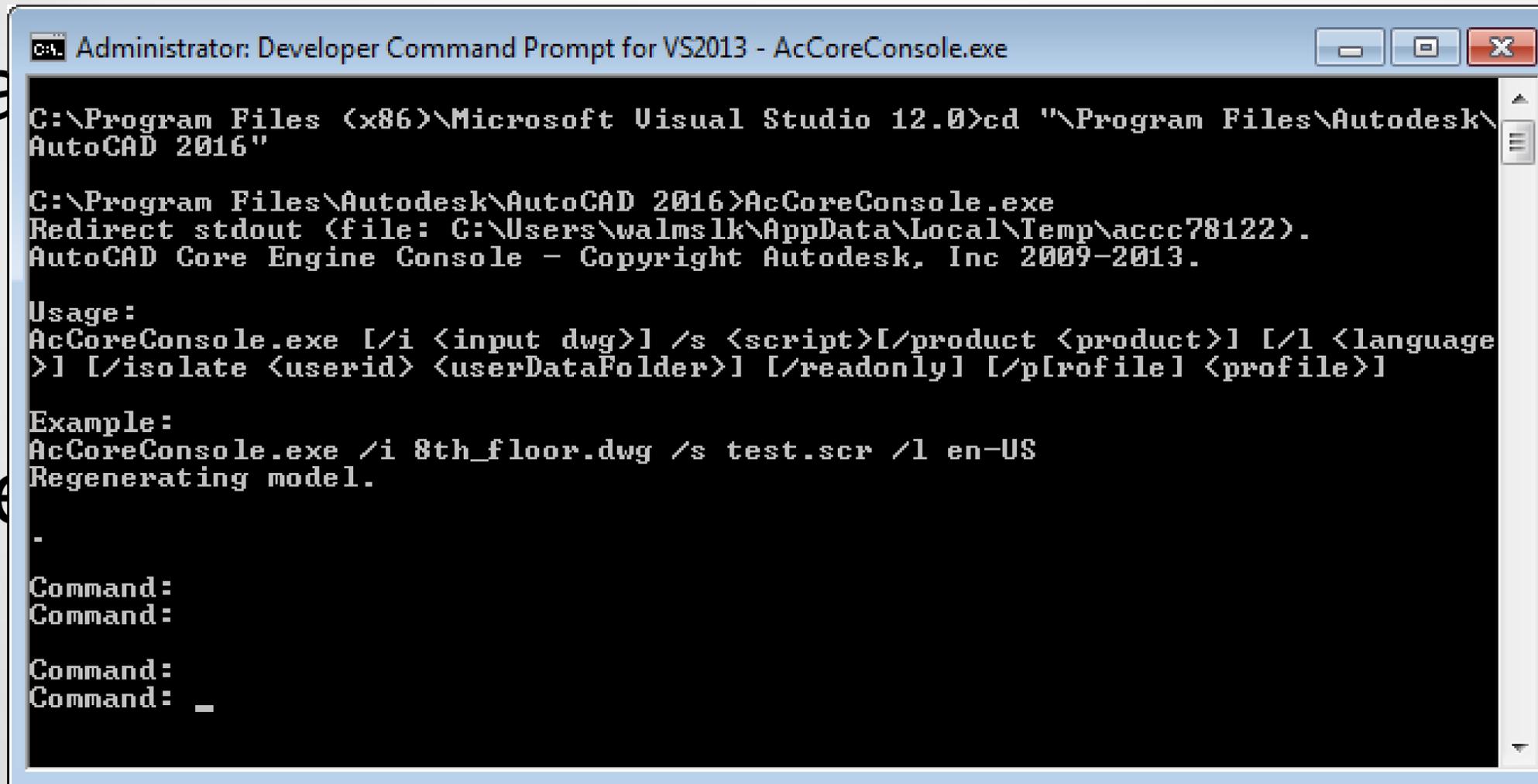
The Core Console

- Headless AutoCAD, running inside a Command Prompt

- Create

- Can

- A gre



```
Administrator: Developer Command Prompt for VS2013 - AcCoreConsole.exe
C:\Program Files (x86)\Microsoft Visual Studio 12.0>cd "%Program Files\Autodesk\AutoCAD 2016"
C:\Program Files\Autodesk\AutoCAD 2016>AcCoreConsole.exe
Redirect stdout (file: C:\Users\walmslk\AppData\Local\Temp\accc78122).
AutoCAD Core Engine Console - Copyright Autodesk, Inc 2009-2013.
Usage:
AcCoreConsole.exe [/i <input dwg>] /s <script>[/product <product>] [/l <language>]
[/isolate <userid> <userDataFolder>] [/readonly] [/p[rofile] <profile>]
Example:
AcCoreConsole.exe /i 8th_floor.dwg /s test.scr /l en-US
Regenerating model.
.
Command:
Command:
Command:
Command: _
```

Running custom AutoCAD code in AutoCAD I/O

- Start by making it work with the Core Console
 - Target *AcDbMgd.dll* and *AcCoreMgd.dll* but not *AcMgd.dll*
 - Command-line/canvas selection UI only
 - `Editor.GetXxx()` functions
- A few changes needed for AutoCAD I/O...
 - Runs with limited local privileges: no direct network access
 - The CoreRunner process downloads any remote content needed
 - Command parameters encoded as (e.g.) JSON
 - Use `Editor.GetString()` to get the location of the JSON file

Creating a custom Activity with a .NET AppPackage

Administering AppPackages and Activities

- Typically use a console app for administration
 - Embeds your credentials locally
 - Never gets distributed to customers
- Simplest approach: clone and modify another sample
 - github.com/Developer-Autodesk/autocad.io-custom-activity-apppackage-Csharp
 - Builds your “CRX app” and uploads it as a Zipped AppPackage
 - Creates a corresponding Activity that makes use of it

Calling a custom Activity from your web-site

You're going to need a web-service

- Your web-page cannot call directly into AutoCAD I/O
 - Embedding of credentials being the primary reason
- Can use lots of server-side technologies for this
 - ASP.NET is a popular choice for existing .NET developers
- Your web-service will need an authorization token
 - Requests it from the authentication API using your credentials
 - Uses it to sign the various AutoCAD I/O requests

Web-service manages WorkItems via HTTP

- Jigsawify.com uses Node.js for server-side JavaScript
 - Same edge detection code as the browser
- Web-service uses POST to create WorkItems
 - <https://developer.autodesk.com/autocad.io/us-east/v2/WorkItems>
 - Authorization headers
 - OData 4.0 payload for parameters
- Checks status using GET
 - Same URL + "(Id=12345678)"

Browser

Node.js

AutoCAD I/O



edgedetector.js



/a /submit?pieces=10&wid
th=12&height=18&unit=inc
he&threshold=10&pixels=
...



**Too much data to
encode as URL
parameters**

Browser

Node.js

AutoCAD I/O

Scaled down image



edgedetector.js



blob1438009040488



/api/submit?pieces=100&width=12&height=18&units=inches&threshold=70&upload=blob1438009040488



Jigsaw.png

Stores image in folder

Larger images caused a server-side problem (with logging)



edgedetector.js

Extracts contents of ZIP

Zippped DWG & PNG

JigsawActivity

Generates puzzle geometry, plots image

Browser

Node.js

AutoCAD I/O

Scaled down image



edgedetector.js



blob1438009040488



/api/submit?pieces=100&width=12&height=18&units=inches&threshold=70&upload=blob1438009040488



Jigsaw.png

Stores image in folder

edgedetector.js

Write engraving to disk

Extracts contents of ZIP

Activities are sandboxed: no direct network access

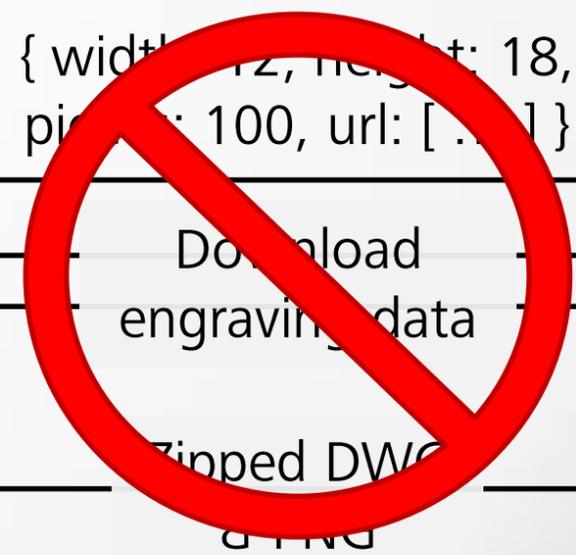
{ width: 12, height: 18, pieces: 100, url: [...] }

Download engraving data

Zippped DWG & PNG

JigsawActivity

Generates puzzle geometry, plots image



Browser

Node.js

AutoCAD I/O

Scaled down image



edgedetector.js



blob1438009040488



/api/submit?pieces=100&width=12&height=18&units=inches&threshold=70&upload=blob1438009040488



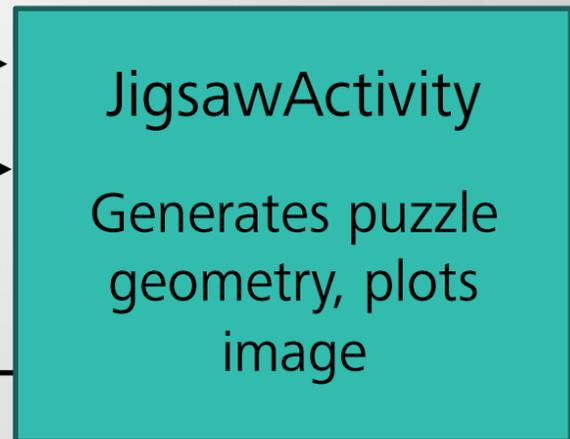
Jigsaw.png



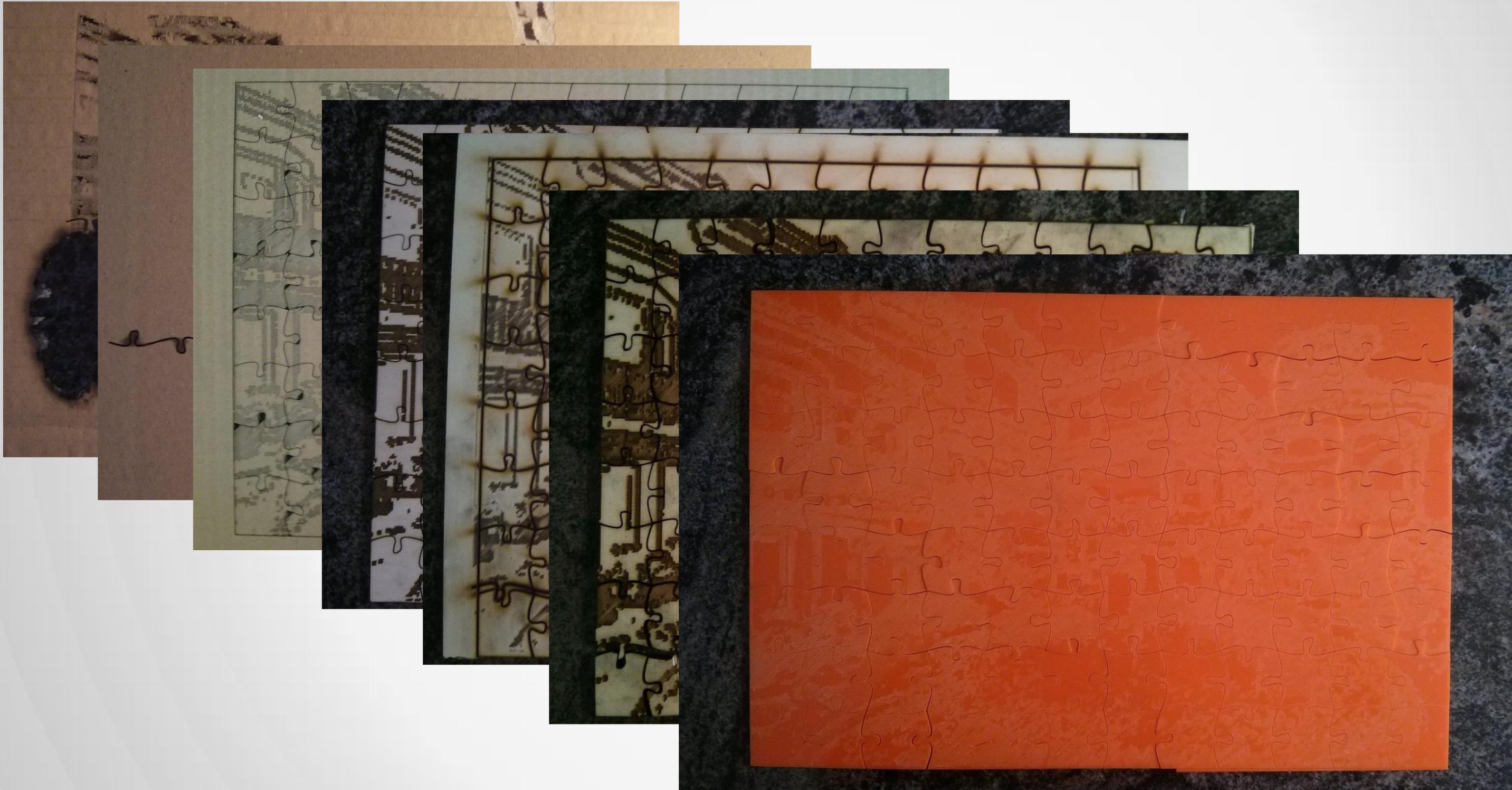
{ width: 12, height: 18, pieces: 100 }

URL to pixels as additional input parameter

Zipped DWG & PNG

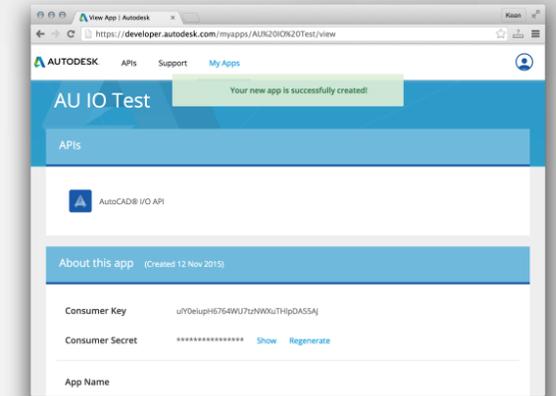
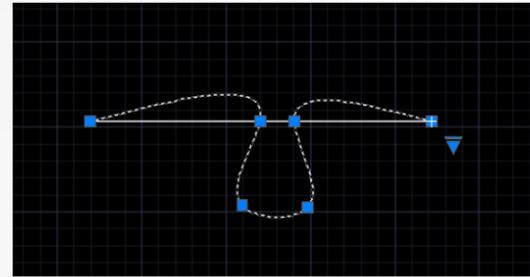


The output



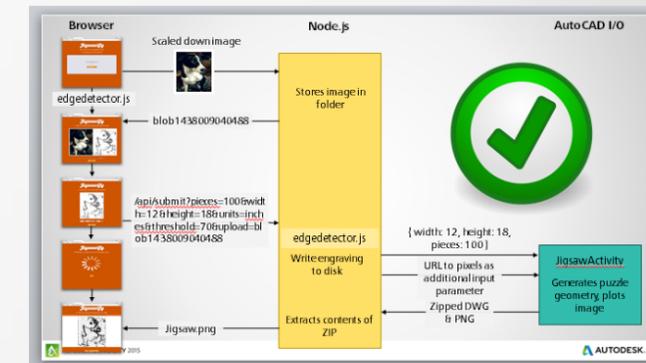
Summary

- The scenario
- Getting started with AutoCAD I/O
- Creating core modules
- Creating & calling custom Activities



```
Administrator: Developer Command Prompt for VS2013 - AcCoreConsole.exe
C:\Program Files (x86)\Microsoft Visual Studio 12.0>cd "%Program Files\Autodesk\AutoCAD 2016"
C:\Program Files\Autodesk\AutoCAD 2016>AcCoreConsole.exe
Redirect stdout (file: C:\Users\malink\AppData\Local\Temp\accc78122).
AutoCAD Core Engine Console - Copyright Autodesk, Inc 2009-2013.
Usage:
AcCoreConsole.exe [/i <input dwg>] [/o <script>[/product <product>] [/l <language>]] [/isolate <userid> <userDataFolder>] [/readonly] [/profile <profile>]
Example:
AcCoreConsole.exe /i 8ch_floor.dwg /o test.scr /l en-US
Regenerating model.

Command:
Command:
Command:
Command: _
```



Join a user research session at AU!

Tell us what you think!

Multiple session times available on
Wednesday and Thursday



Get an Amazon Gift
Card for joining!

Web Developers, Unite!

90 minute Group Session

How are you using the Web APIs?

What APIs would you like to see next?

Share your work and hear what others
are already doing with Web APIs



To sign up, scan the QR code or visit: **autode.sk/webdev**

Be heard! Provide AU session feedback.

- Via the Survey Stations, email or mobile device.
- AU 2016 passes awarded daily!
- Give your feedback after each session.
- Give instructors feedback in real-time.



More Questions? Visit the AU Answer Bar

- Seek answers to all of your technical product questions by visiting the **Answer Bar**.
- Open daily 8am-10am and Noon-6pm and located just outside of Hall C on Level 2.
- Staffed by Autodesk developers, QA, & support engineers ready to help you through your most challenging technical questions.



