

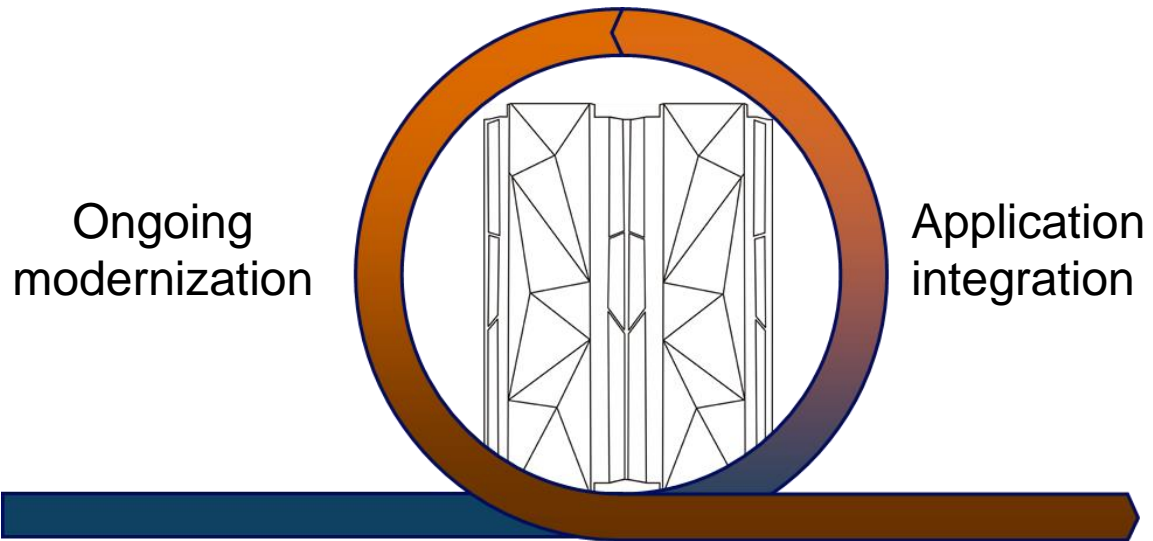
# Optimise your Batch with Application Discovery

David Baxandall  
IBM

November 2018  
Session NF



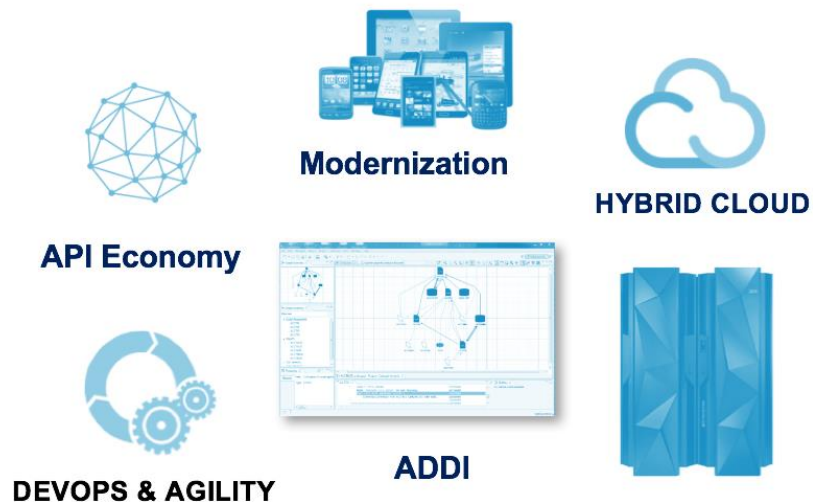
# Why automated application understanding?



- Mainframe applications were originally architected for speed and CPU overhead – not ease of maintenance or change
- Business requirements such as Digital Transformation are building on existing application assets and driving modernization
- Accelerated drive towards modern development practices to reduce cost and shorten delivery cycles
- End to end understanding requires cross platform analysis, not just z/OS

# Application Discovery and Delivery Intelligence (ADDI)

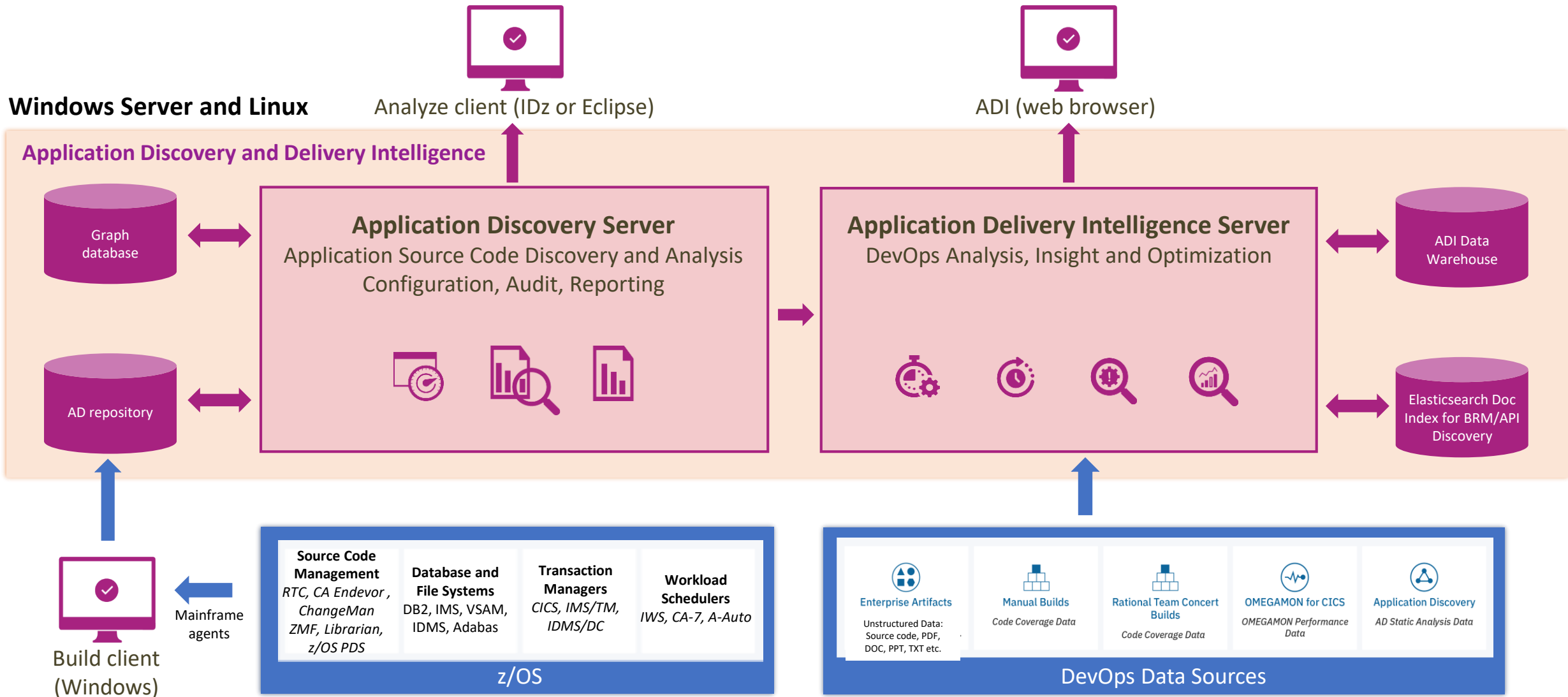
*Quickly and safely unlock the value of existing assets for digital transformation*



**Reduce risk, increase productivity,  
and accelerate your digital transformation**

- Rapidly analyze and visualize relationships between application components, data, and jobs to make changes safely and efficiently
- Reduce risk by reusing current assets and understanding impacts of potential changes
- Increase productivity by automating interactive documentation, enabling new workers to be effective in a fraction of the time, and optimizing tests
- Accelerate your digital transformation by rapidly identifying candidates and ensuring integrity of APIs

# IBM ADDI Architecture



# Schedulers - summary

- Customers will usually have an incomplete view of their combined batch environment as for on-line applications
- The batch environment comprises schedules, JCL, files, databases and programs – all inter-related to the on-line environment. **IWS displays schedules and JCL steps – AD provides wider analysis for additional components and drill down**
- The difficulties, costs and resource overhead in conducting manual analysis is the same as for on-line development / maintenance
- Digital modernization also affect batch and necessitates change
- ADDI is the **ONLY** platform to analyse and correlate all batch and on-line components

Note: In IWS terminology an ‘application’ is a collection of JCL’s

# ADDI IBM Workload Scheduler analysis

## Why analyse batch schedules?

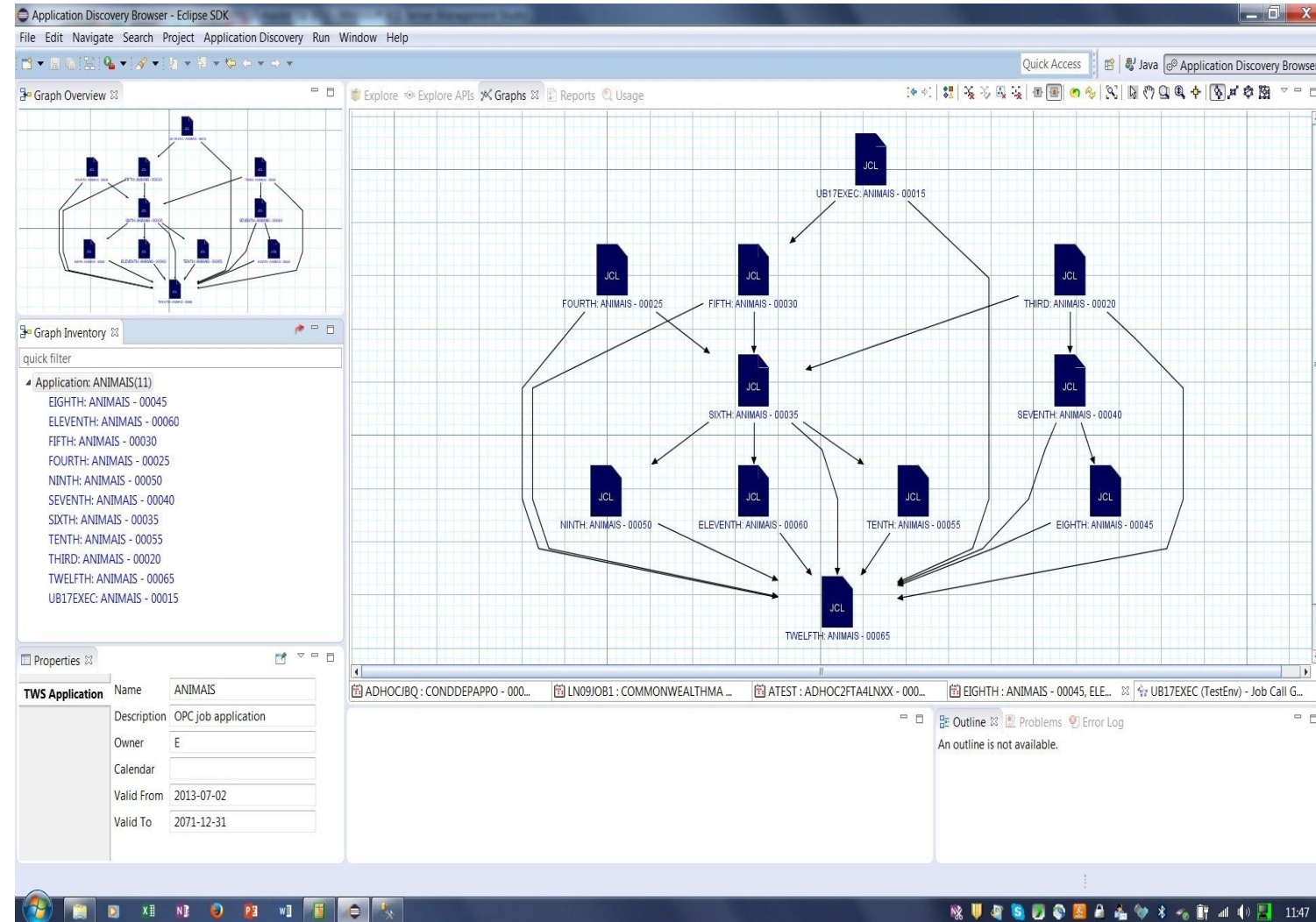
- Batch is a significant proportion of mainframe applications and is time consuming to understand manually
- Interaction between the batch and online environments is often poorly understood and so change and testing activities are more complex and costly than necessary
- Modernization and digital modernization is leading to an increased rate of change
- Reduce the reliance on the availability of SME's for all change requests
- Reduce the costs and time for maintenance, change and test activities



# Batch Schedule Analysis

## Process:

1. Mainframe Agents use iWS API EQQYCON to retrieve and download Schedule Information, which is then loaded directly to the underlying repository.
2. AD Analyze client graphically renders the Schedule Information.
3. The Graph shows the sequence and dependencies of the jobs associated with selected schedule(s) or applications.
4. Filtering operations are available to show specific paths within the overall schedule flow.



iWS Schedule Graph

# Batch Schedule Analysis

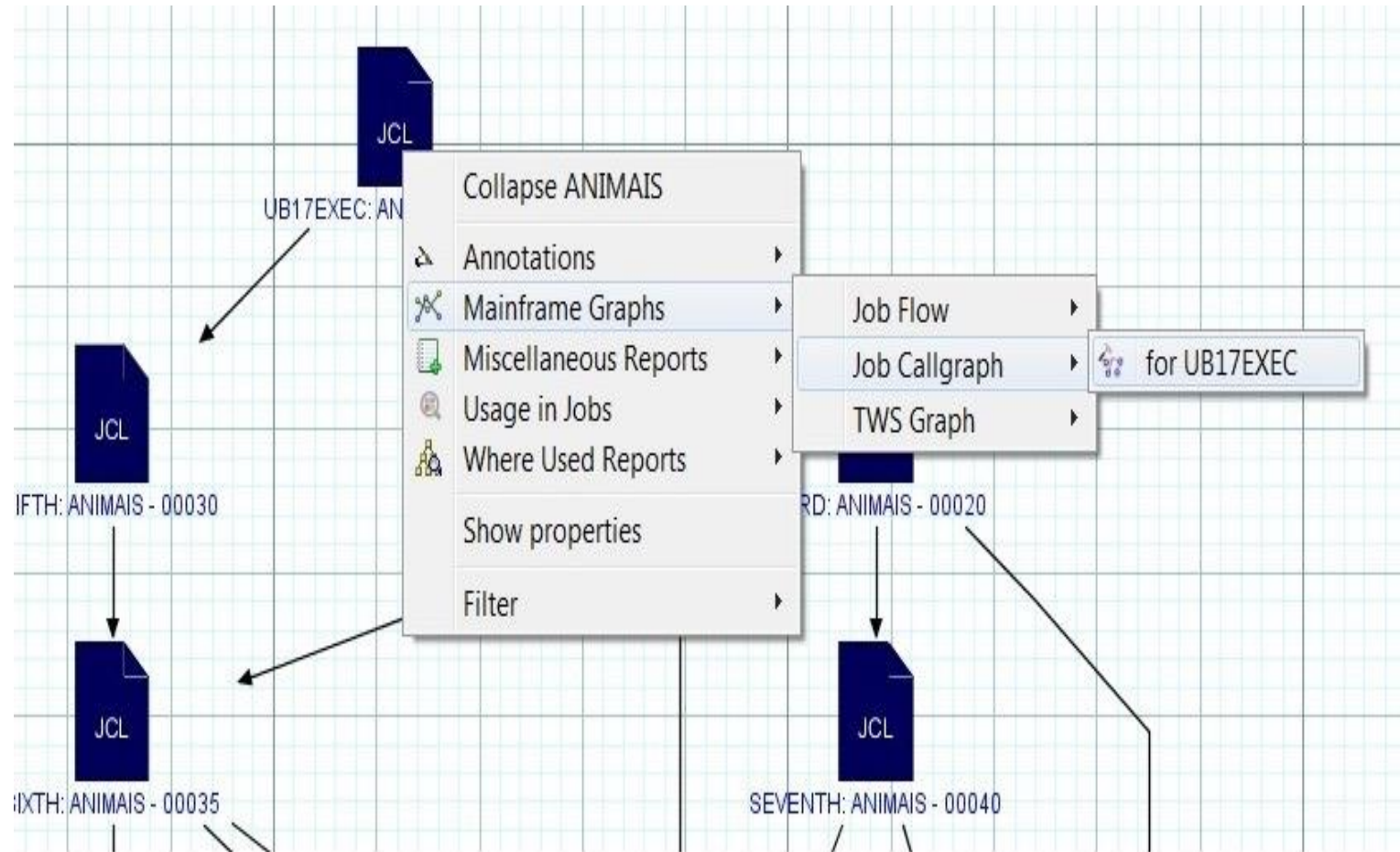
Process:

5. From the Jobs shown in the Batch Schedule, you may then invoke appropriate functionality

6. Drill down from the Schedule to the lower levels of Job and Program analysis, including:

- Program Call Chains
- File and Database I/O
- Program Structure Graphs
- Statement-level flow chart

7. Inclusion of Schedule information allows a holistic view of the Batch environment.



Invoking Job Callgraph from iWS Graph



# Demonstration

# We want your feedback!

- Please submit your feedback online at ....
  - <http://conferences.gse.org.uk/2018/feedback/NF>
- Paper feedback forms are also available from the Chair person
- This session is NF

