# Decomposition and Orchestration

with SAS Data Integration Studio
and LSF

SAS User Group UK & Ireland – 2018-12-06

# decomposition

/ˌdiːkɒmpəˈzɪʃn/

*noun*

the state or process of rotting; decay.
"the decomposition of organic waste"
*synonyms:* decay, rotting, going bad, putrefaction, putrescence, putridity, festering, spoilage, perishing; More

# orchestration

/ɔːkɪˈstreɪʃ(ə)n/

*noun*

1. the arrangement or scoring of music for orchestral performance.
   "Prokofiev's mastery of orchestration"
2. the planning or coordination of the elements of a situation to produce a desired effect, especially surreptitiously.
   "the orchestration of the campaign needed tightening"

# What am here tonight to talk about?

## Mindset

**not about the tools you use,**
**but the way you look at the problem.**

## Tools

**the way you look at the problem,**
**drives the way you use the tools.**

# How you look at **the problem**

- create, operate and sustain large-scale complex data ingest processes
- a range of development, test, operate and maintain life-cycles
- large teams of developers, administrators and operators
  - with a range of skill and experience levels, roles and responsibilities
- promote reuse and maintainability
- service end-to-end data lineage demands
- avoid costly to sustain code-only approaches

# drives the way you use **the tools**

**Data Integration Studio**

- build ingest processes – “jobs”
- deploy jobs as SAS programs

**IBM Platform LSF**

- “scheduling” system to execute the jobs built with DI Studio
- much, much more importantly **orchestrates** the execution of the jobs

# how the tools work informs and enables the approach

**Data Integration Studio**

in no way imposes any particular approach to how you assemble, or disassemble, the steps in you processes

smallest unit of processing - job

**LSF (IBM Platform LSF)**

“scheduling” system to execute the jobs built with DI Studio

much, much more importantly **orchestrates** the execution of the jobs

# decomposition and orchestration - what do we mean

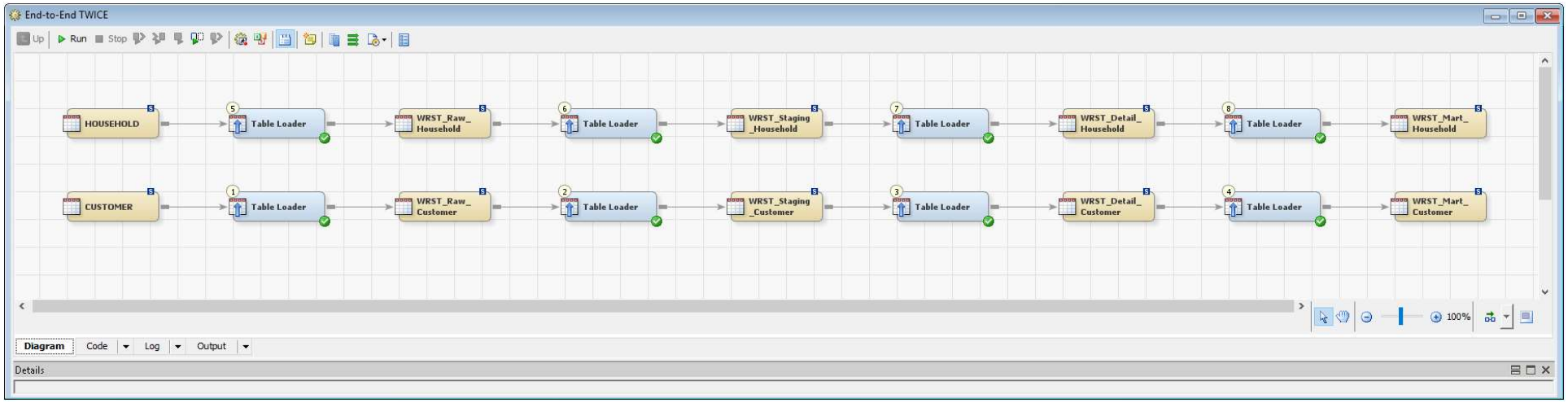
## decomposition

breaking a complex problem or system into parts that are easier to conceive, understand, program, and maintain

## orchestration

co-ordinating operation of the component parts of system or process to deliver the its objectives

# a worked example

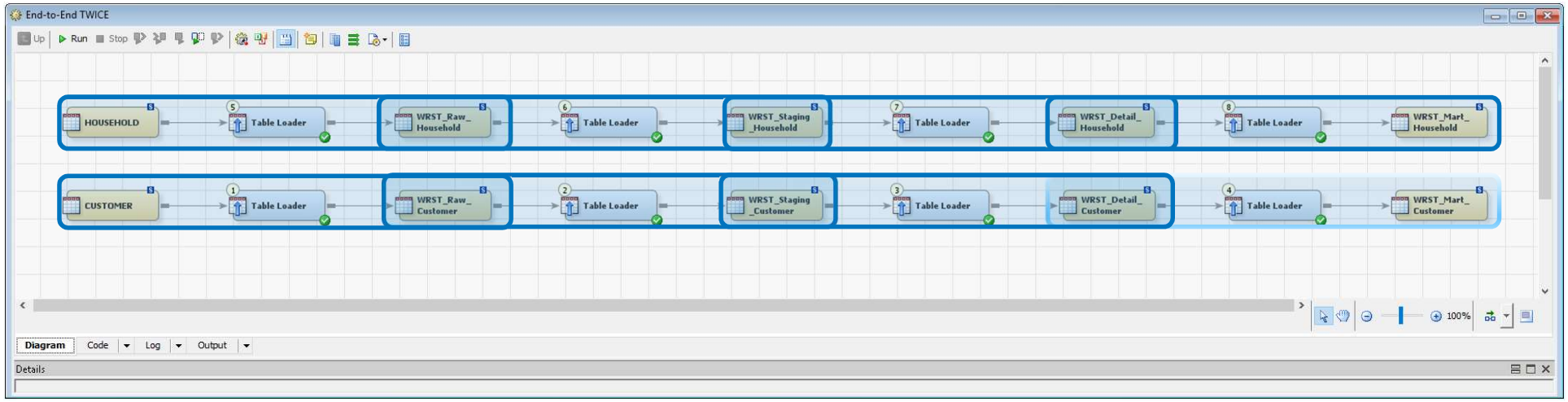



## end-to-end, twice

two completely independent processing threads in one job

each thread comprises four steps

# a worked example




**decomposes into eight self-contained steps**

smallest possible unit of processing, in this case

two threads comprising four jobs

each job deployed as “deployed job” (and SAS program file)

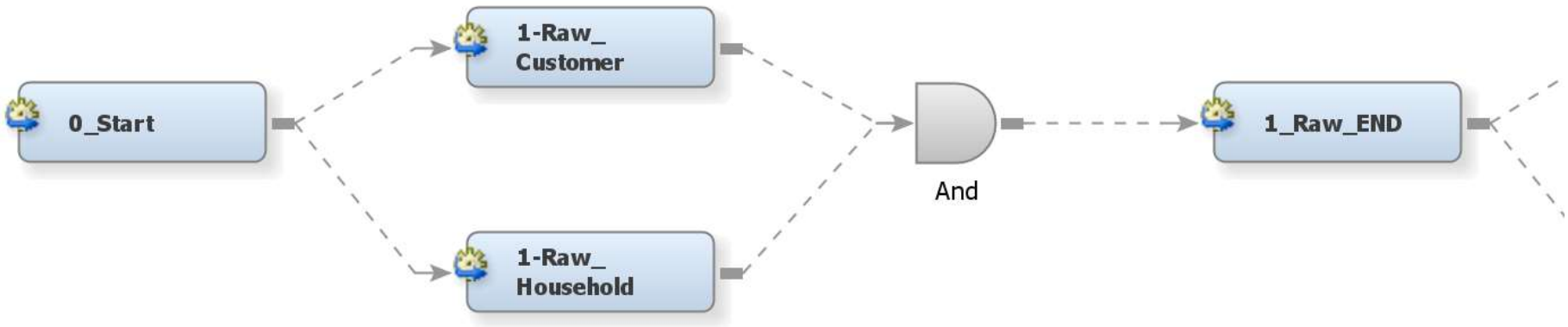
# a worked example


**deployed jobs are assembled into a flow**

Schedule Manager in Management Console

then Scheduled (“deployed”) to LSF for execution

# a worked example




**deployed jobs are assembled into a flow**

Schedule Manager in Management Console

then Scheduled (“deployed”) to LSF for execution

# how to decompose

## some principles

**idempotence**

jobs can be re-run and produce the same result

inputs are inviolate

if there are assumptions, enforce them

**dependence**

sequence is the responsibility of orchestration

process error handling is the responsibility of orchestration

**balance**

decomposition is not a goal in itself

context is important but is likely to change over time

# mindset

**orchestration capabilities are intrinsic to approach**

**decompose because you can orchestrate**

DI Studio encourages breaking processing into tangible steps -

orchestration should encourage breaking whole end-to-end processes into steps

**parallelism**

decomposed independent jobs will execute in parallel

decomposition is also about performance – as compared to monolithic approaches

**logical vs practical**

logically decomposed workflows may be impractical

orchestration framework will have features to address this

meta-scheduling also relevant – but that's a topic for another day