A distributed Optimization Bot/Agent Application Framework for GAMS Models

Franz Nelißen – FNelissen@gams.com

Implementation of Prototype by Girish Garg

Agenda

Application Building for AML

A Prototype

Summary and Outlook

Application Building for AML

Some Approaches

Integration with Analytical Software

- "Top-Down" - Add AML to existing analytical software systems with "large" user base, e.g. MATLAB or SAS
- "Bottom-Up" - Add GUI-builder / Application Framework to AML with "small" user base, e.g.: AIMMS (Pro) or FICO Xpress-Insight

Application Building for AML

Some Approaches

Integration with Programming Language

- "Top-Down"- Extend existing programming language with declarative AML, e.g. : Pyomo (Python), JuMP (Julia), MS Solver Foundation (discontinued)
- "Bottom-Up" - Make it easy to embed GAMS into different (programming) environments

Seamless Integration

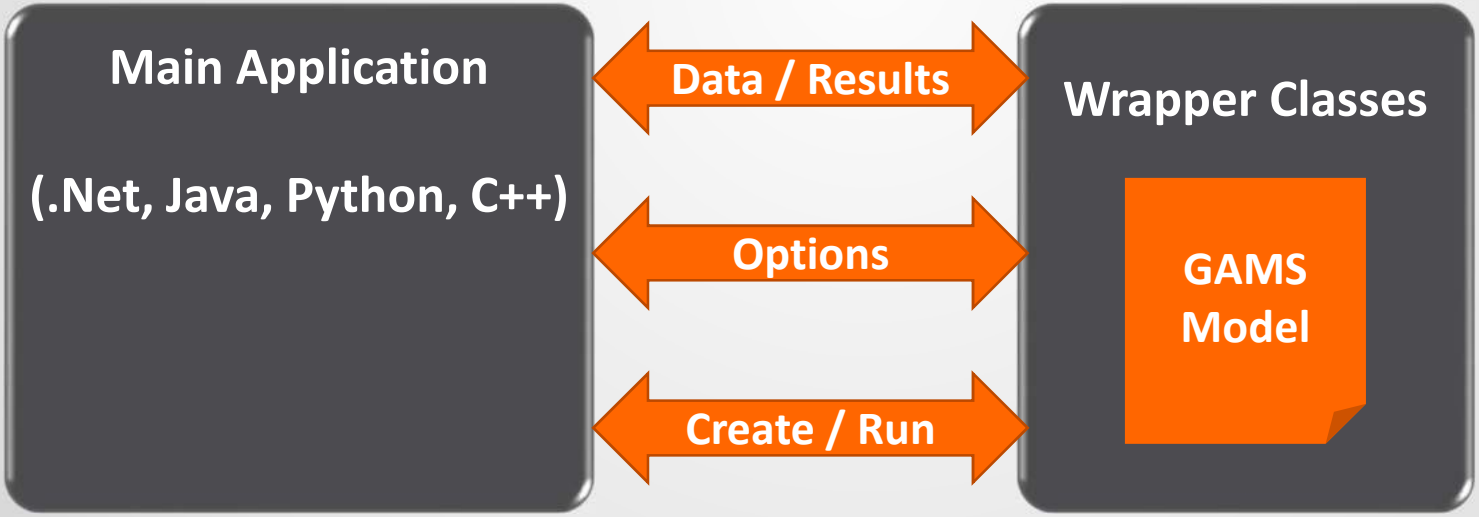
Separation of Tasks

- Use GAMS for modeling and optimization tasks
- Object oriented GAMS API connects GAMS to other environments to build Applications
  - Programming languages
  - Smart Links to Databases, Spreadsheets, Matlab, R,…
- .Net, Java, Python, C++ (open source)
- Communication through Memory or Files

OO-API: Encapsulation of GAMS Model

Simple Interface to interact with GAMS

- Classes to communicate input data and results
- Classes to change options like the solver to use
- Classes to create, run, and control model instance(s)

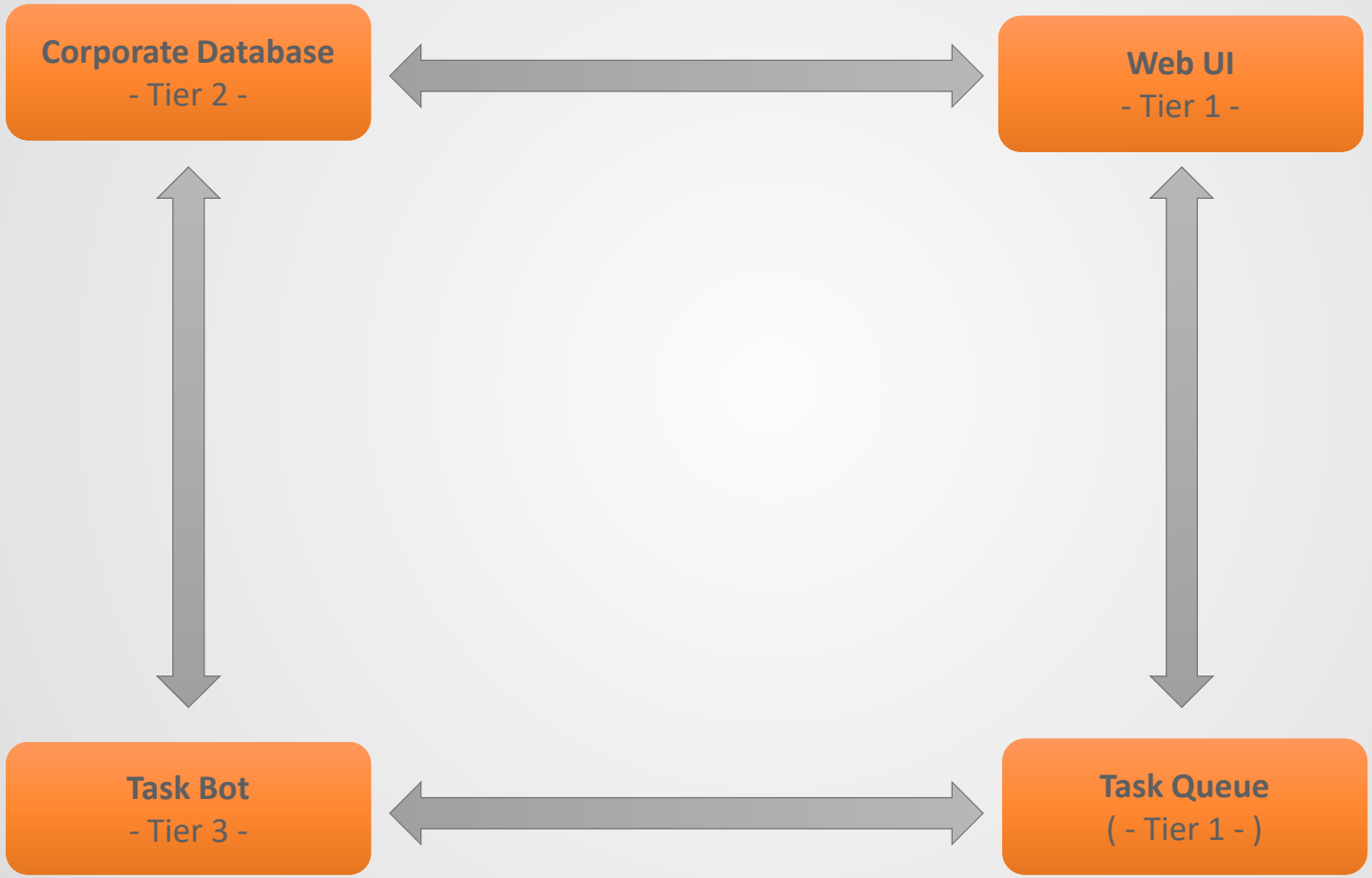


Task At Hand

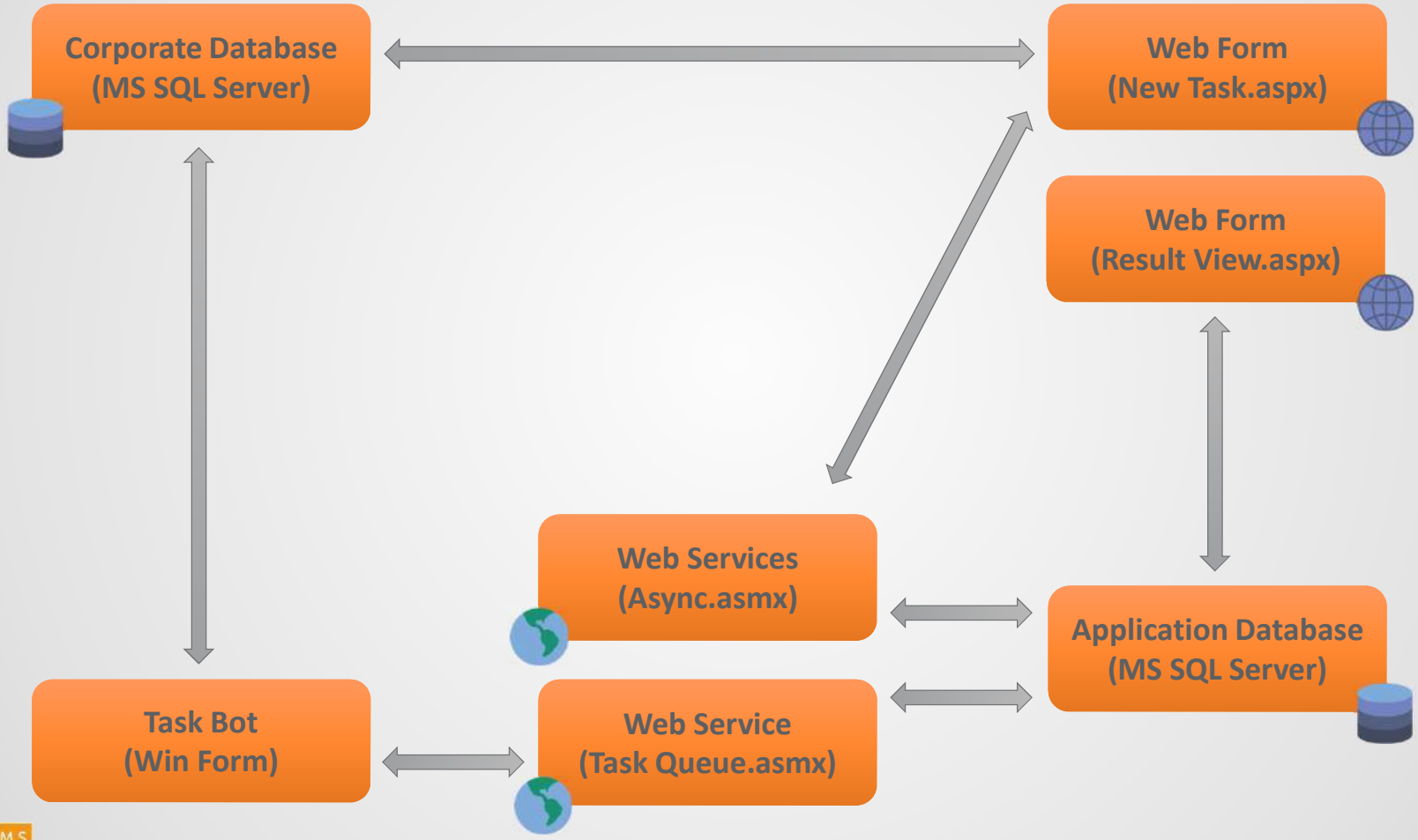
Develop Prototype of a distributed Multi-Tier Application with a Multi-User Web Interface

- Application connects GAMS Model to Databases and Web User Interface
- Bot/Agents run Model instances
- User Interface allows
  - Setup and submission of (multiple) GAMS jobs
  - Visualization of results
- Communication with GAMS through OO-API only
- .Net Application
- Application Developer has no knowledge about GAMS
- Tight Budget for Application Development

Architectural Overview



Technical Overview



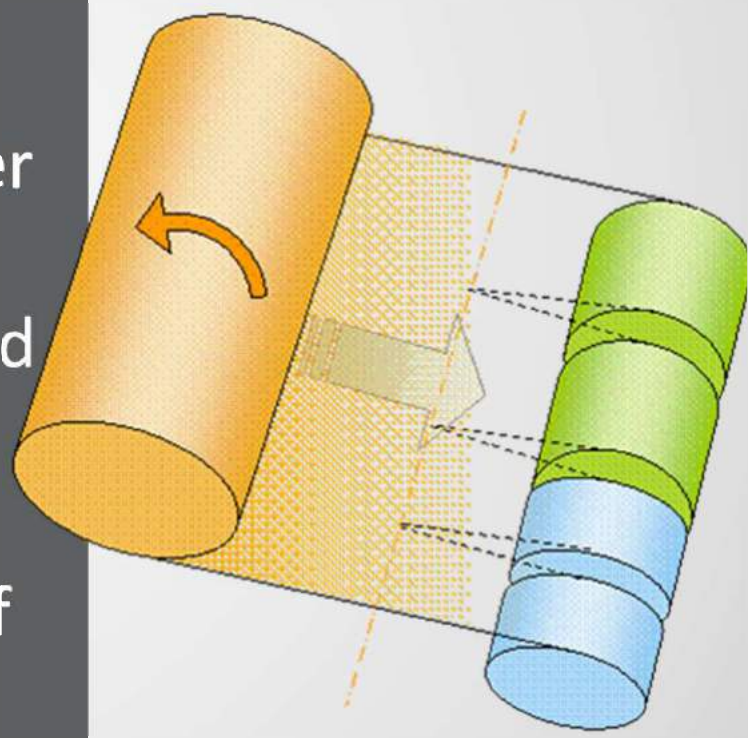
Some Features

- Asynchronous architecture: Task submission and execution are decoupled through the Task Queue
- Data Contract between Bots and Application: Common Data Structures for clear interface
- Distributed system (multiple tiers): WebUI, Databases and Bots can run on separate machines
- Scalable with multiple bot instances

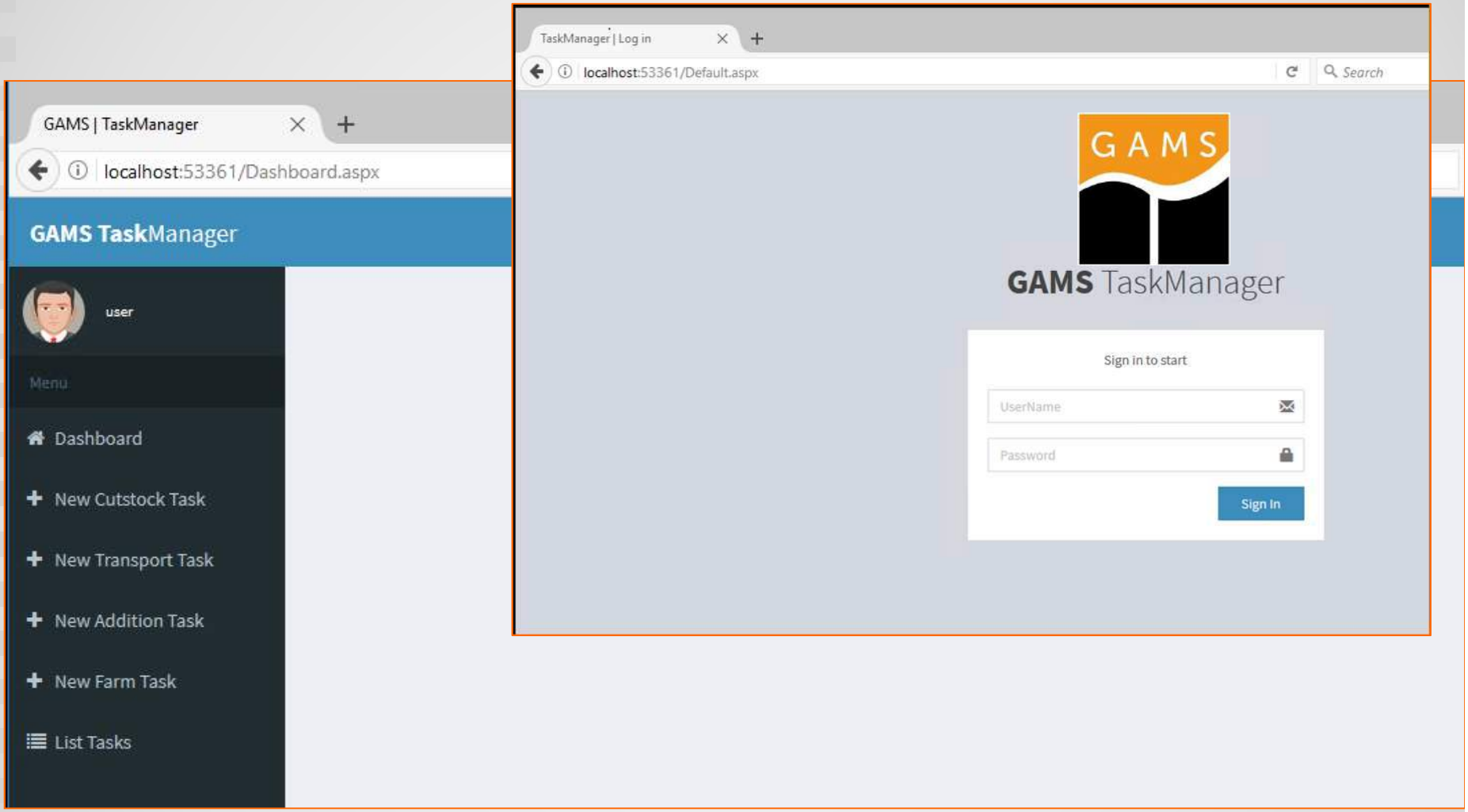
Example: Roll Cutting (cutstock.gms)

Cut paper rolls of fixed width ("raw") into smaller portions ("finals")

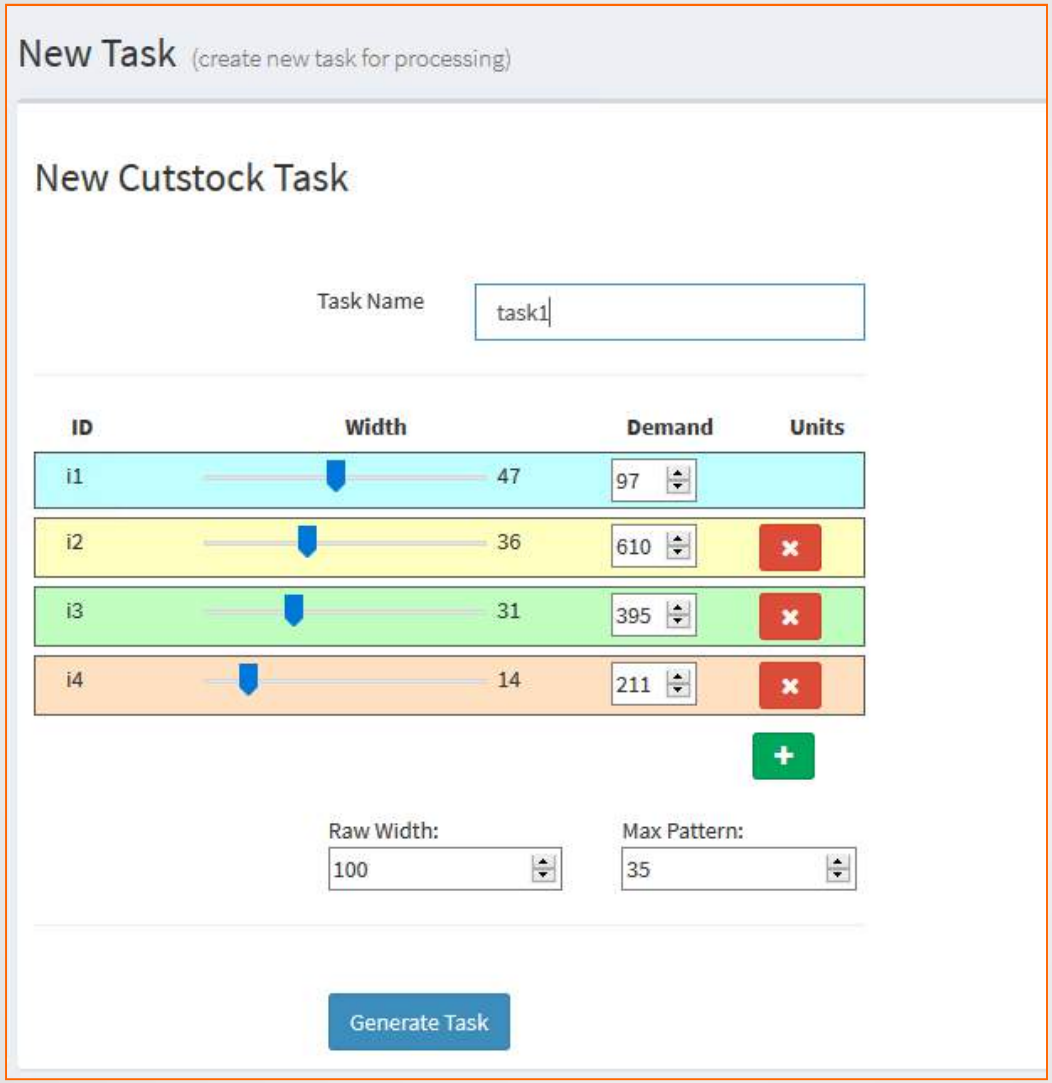
- Input:
  - Width of the raw
  - Demand: Widths and number of finals
- Objective: Minimize the required number of raws
- Output:
  - Combination and number of cuts ("patterns")
  - Number of required raws



Web User Interface



Create New Cutstock Task



Task List (= Queue) and Status

GAMS TaskManager Change Passw

user

Menu

- Dashboard
- New Cutstock Task
- New Transport Task
- New Addition Task
- New Farm Task
- List Tasks

Task List (previously generated tasks in system)

Hom

Type	Name	Status	Issued On	Allotted On	Completed On	Result
cutstock	task7	available	19-Jul-17 07:59:55	NA	NA	Status Checker
cutstock	task3	completed	18-Jul-17 02:14:14	18-Jul-17 02:14:22	18-Jul-17 02:14:24	View Result
cutstock	task2	completed	18-Jul-17 02:13:58	18-Jul-17 02:14:02	18-Jul-17 02:14:03	View Result
cutstock	task1	completed	18-Jul-17 02:13:45	18-Jul-17 02:13:51	18-Jul-17 02:13:52	View Result
cutstock	task4	completed	18-Jul-17 02:12:04	18-Jul-17 02:12:10	18-Jul-17 02:12:12	View Result
cutstock	task3	completed	18-Jul-17 02:09:41	18-Jul-17 02:10:19	18-Jul-17 02:10:21	View Result
cutstock	task2	completed	18-Jul-17 02:09:28	18-Jul-17 02:10:08	18-Jul-17 02:10:10	View Result
cutstock	task1	completed	18-Jul-17 02:09:16	18-Jul-17 02:10:07	18-Jul-17 02:10:09	View Result

Tasks Results Page (1)

Task Results (task3)

Home > Tasks >

Name	task3	**IssuedOn**	18-Jul-17 02:14:14
Type	cutstock	**AllottedOn**	18-Jul-17 02:14:22
Status	completed	**CompletedOn**	18-Jul-17 02:14:24
Done By	CutstockBot01		

Input

RawWidth	MaxPattern
125	35

Piece Name	Width	Demand	Color
i1	47	97	
i2	36	610	
i3	31	395	
i4	14	211	

Tasks Results Page (2)

Output

pattern 5 : **87** times

i3	i3	i3	i3	E

pattern 6 : **203** times

i2	i2	i2	i4	E

pattern 9 : **47** times

i1	i1	i3	E

pattern 11 : **3** times

i1	i2	i4	i4	i4	E

Log

New pattern! Value: -3

New pattern! Value: -2.33333333333333

New pattern! Value: -1.66666666666667

New pattern! Value: -1.25

New pattern! Value: -0.25

New pattern! Value: -0.041666666666667

New pattern! Value: -0.0220588235294117

Optimal Solution: 340

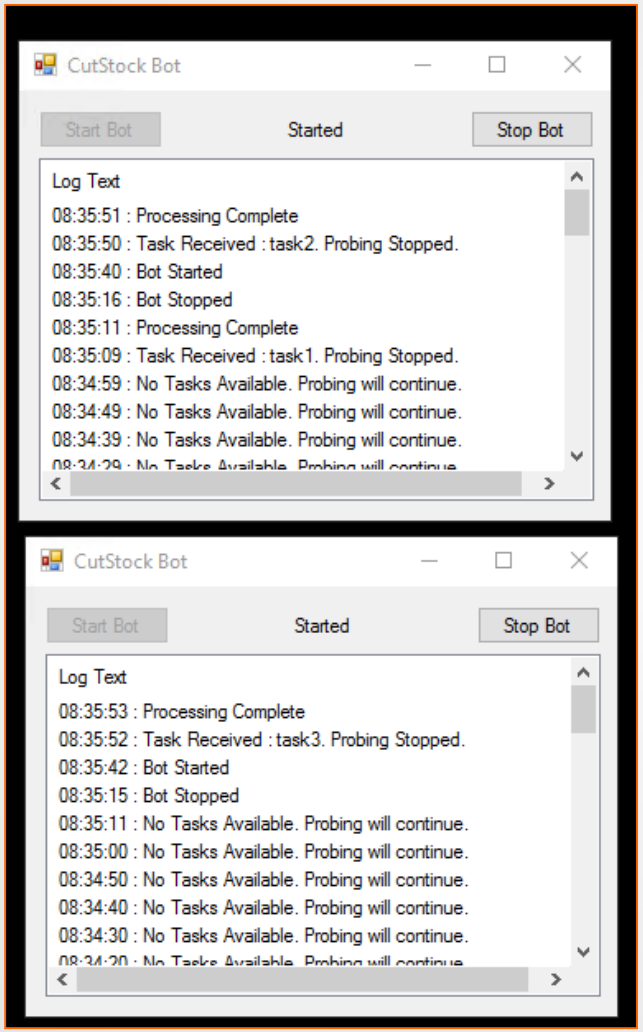
pattern 5 87 times: i3: 4

pattern 6 203 times: i2: 3 i4: 1

pattern 9 47 times: i1: 2 i3: 1

pattern 11 3 times: i1: 1 i2: 1 i4: 3

Cutstock Bots Log



Summary and Outlook

- Building Optimization Applications may be challenging

- GAMS has no preference for a specific User Interface
- OO-API makes it easy to embed GAMS models

- Optimization Bot/Agent Application Framework
  - Integrates Web UI, GAMS, task bots and queues, and databases
  - Distributed system / multiple bot instances
  - Prototype done in .Net using the .Net OO-API
  - Source code for prototype will become open source

Thank You