

COMPUTEROME 2.0

USERS WORKSHOP

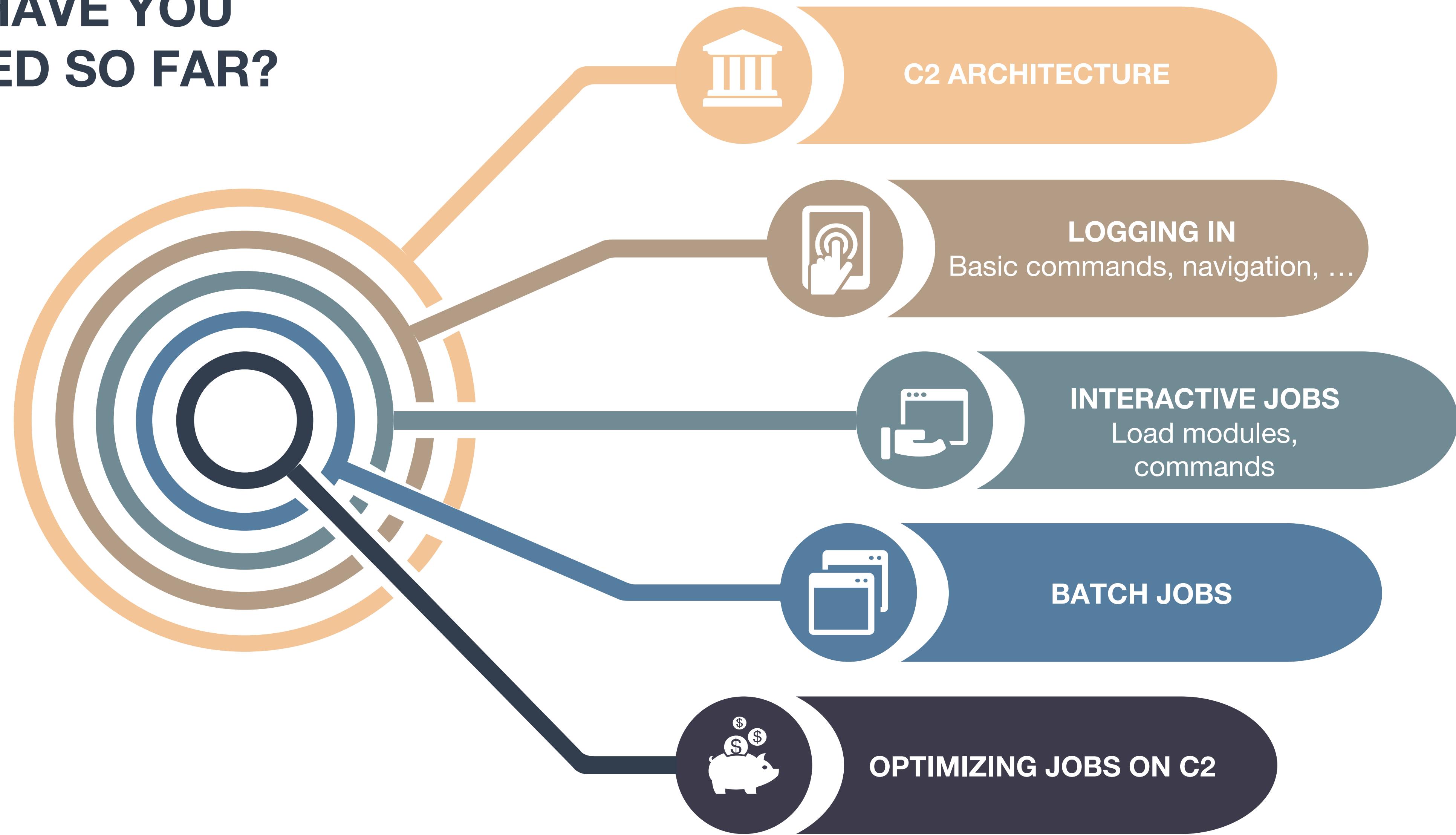
CENTER FOR HEALTH DATA SCIENCE (HEADS)
FACULTY OF HEALTH AND MEDICAL SCIENCES,
UNIVERSITY OF COPENHAGEN, FEBRUARY, 2021



PART5

QUEUEING SYSTEM AND BATCHES

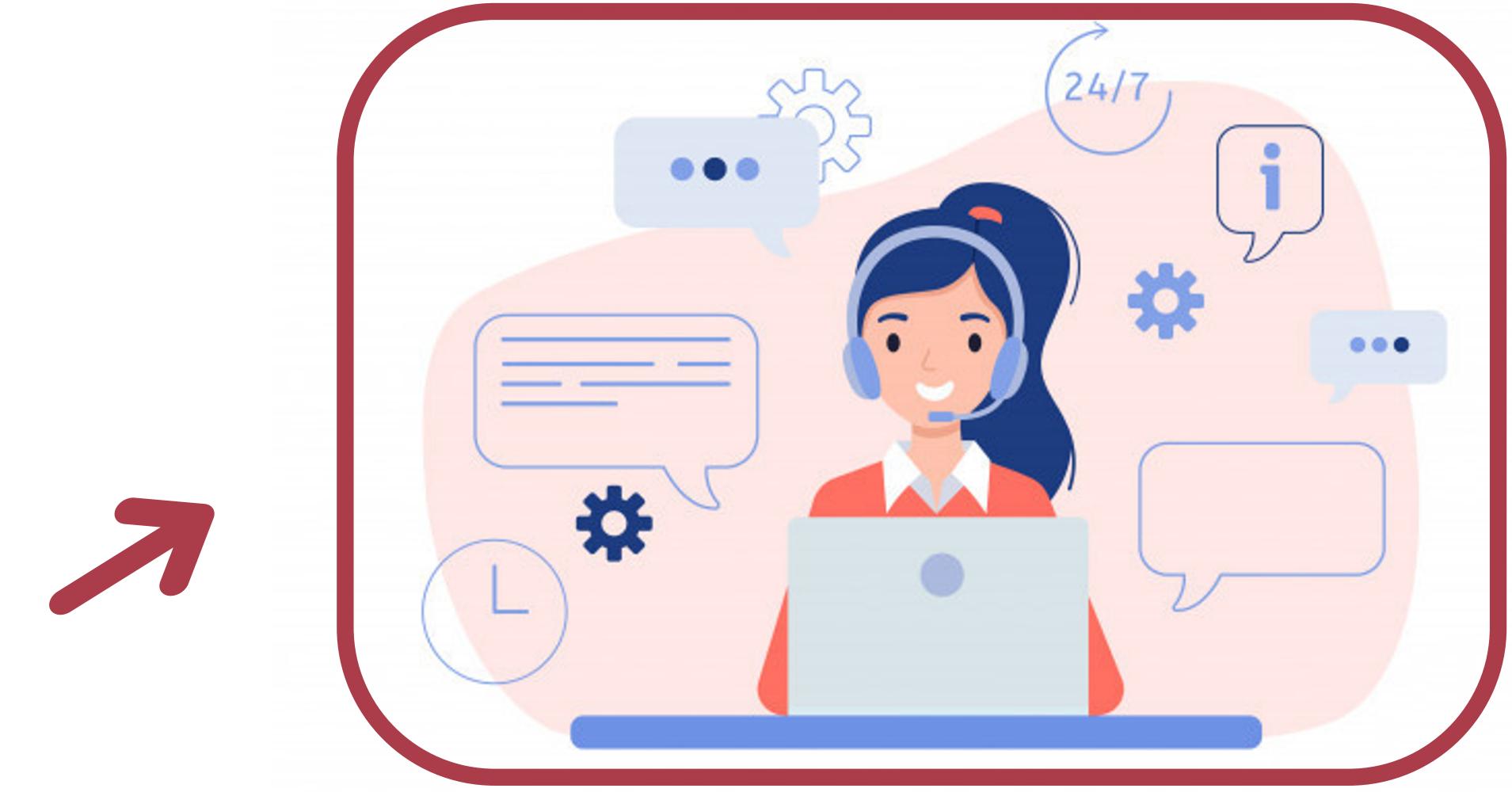
WHAT HAVE YOU LEARNED SO FAR?



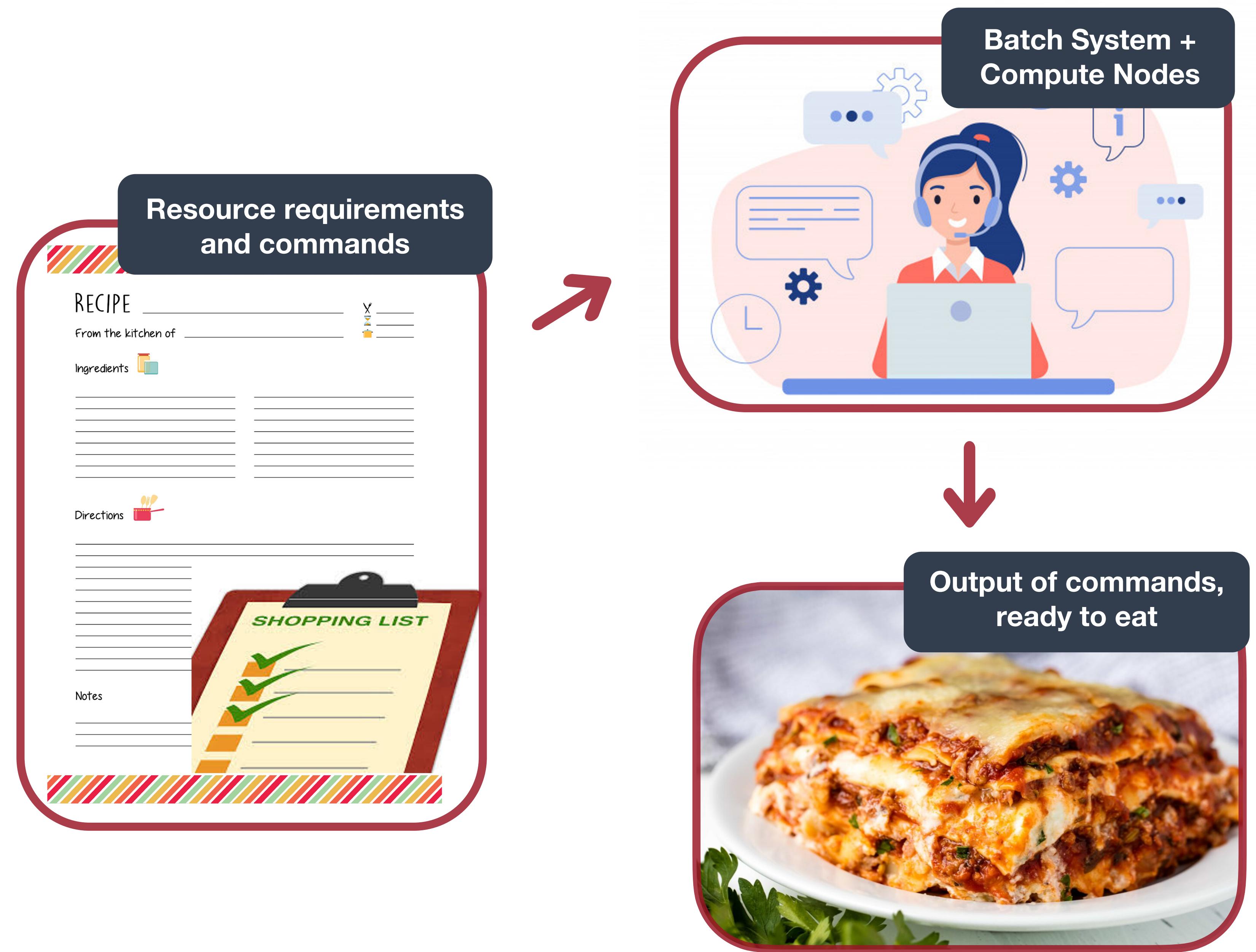
BATCH JOBS

- Batch jobs are resource provisions that run applications on nodes away from the user and **do not require supervision or interaction.**
- Batch jobs are commonly used for **applications that run for long periods of time or require little to no user input.**
- Batch jobs are created from a job script which provide resource requirements and commands for the job.

BATCH JOBS



BATCH JOBS



INTERACTIVE JOBS VS BATCH JOBS

Interactive Job:

- One at a time*
- Need to keep track of job
- Cannot do other things*



Batch Job:

- Many at a time
- Batch system can keep track of jobs
- Live life to the fullest



HOW DO I RUN BATCH JOBS? - qsub command

... a job script which provide resource requirements and commands for the job ...

Running a batch job is easy:

```
qsub [options] myScript.sh
```

TEMPLATE SCRIPT

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n

### Load modules
module load xxx/version
module load yyy/version

### Run your jobs
Job1
Job2
...
```

TEMPLATE SCRIPT

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n

### Load modules
module load xxx/version
module load yyy/version

### Run your jobs
Job1
Job2
...
```

SHEBANG LINE

BATCH SYSTEM

SPECIFICATIONS,
RESOURCES, ...

MODULE LOADS

COMMANDS

!

BATCH SYSTEM SPECIFICATIONS

Let us break it down:

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```



Which account should
be charged for the job?
Required!

BATCH SYSTEM SPECIFICATIONS

Let us break it down:

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```



Name of the job. Optional - by default the name of the script.

BATCH SYSTEM SPECIFICATIONS

Let us break it down:

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

Error and output files:
Optional- if not specified,
derived from job name.
Error: jobname.ejob_id
Output: jobname.ojob_id

BATCH SYSTEM SPECIFICATIONS

Let us break it down:

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

Resource requirements
Multiple different resources specified on the same line.
Required!

BATCH SYSTEM SPECIFICATIONS

Let us break it down:

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

Three resources have to be specified for a job to run on C2:

1. Number of nodes and cores per node:

nodes=1:ppn=40

1 node and 40 cores

1. Amount of memory:

mem=50gb

50 GB of RAM

1. Time allotment for the job:

walltime=1:00:00

1 hour (dd:hh:mm:ss)

BATCH SYSTEM SPECIFICATIONS

Let us break it down:

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory →
### Mail on failure
#PBS -m n
```

Working directory
where the job should
be run.
Optional - by default
your home directory.

BATCH SYSTEM SPECIFICATIONS

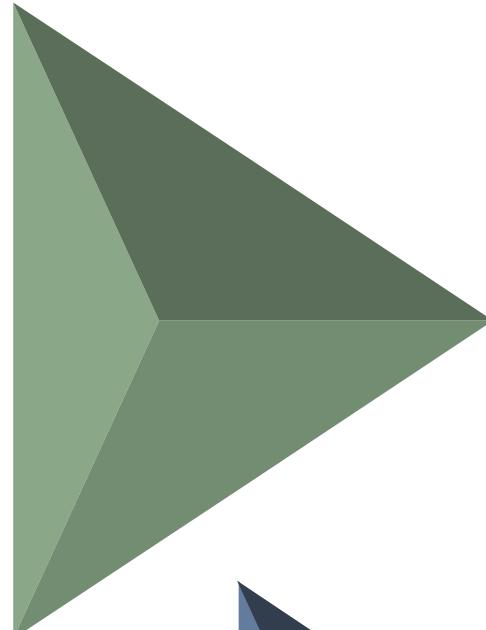
Let us break it down:

```
#!/bin/bash

### Account information
#PBS -W group_list=ku_fa -A ku_fa
### Job name
#PBS -N jobName
### Error stream
#PBS -e myError.err
### Output stream
#PBS -o myOut.log
### Resource requirements
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
### Working directory
#PBS -d workingDirectory
### Mail on failure
#PBS -m n
```

**Get email on job status.
Optional.**

MORE BATCH SYSTEM OPTIONS



#PBS -j oe

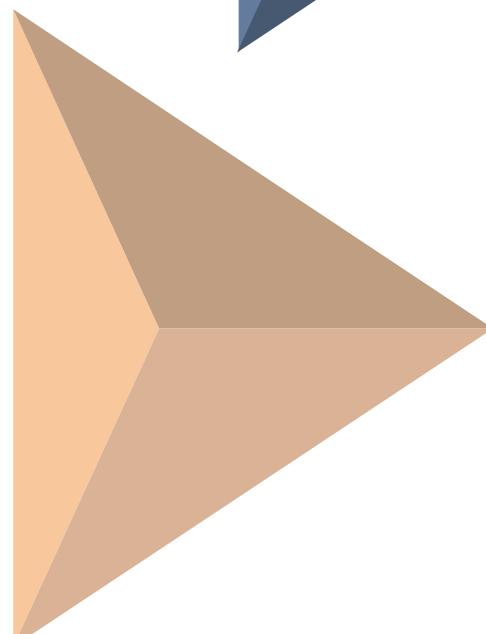
Merge the output and error stream into one stream - the output stream.



#PBS -k oe

Merge the output and error stream into one stream - the output stream

- **with continuously updated streams in your home directory.**



#PBS -V

Transfer environmental variables to compute node

- **NOT RECOMMENDED - DO NOT USE**

EXAMPLE SCRIPT

```
#! /bin/bash
#PBS -W group_list=ku_fa -A ku_fa
#PBS -N canid1Map
#PBS -e canid1.err
#PBS -o canid1.log
#PBS -l nodes=1:ppn=40,mem=50gb,walltime=1:00:00
#PBS -d /home/projects/C2_test

### Load modules
module load bwa/0.7.15
module load samtools/1.9
module load htslib/1.9

### Run your jobs
bwa mem -t 40 refgenome/canFam31.fasta Canid1.R1.fastq.gz > Canid1.sam
samtools view -b Canid1.sam > Canid1.bam
samtools sort Canid1.bam > Canid1.sort.bam
samtools index Canid1.sort.bam
```

| WHAT ORDER DO THE JOBS GET LAUNCHED IN?

HOW DO I MONITOR MY JOBS?

```
shygop@g-12-10002 ~/projects/C2_test
$ checkjob 30518436
job 30518436

AName: canid1Map
State: Completed
Completion Code: 0  Time: Wed Feb 17 15:19:23
Creds: user:shygop group:ku-cbd account:ku-cbd class:batch
WallTime: 00:01:10 of 1:00:00
SubmitTime: Wed Feb 17 15:18:05
(Time Queued Total: 00:00:08 Eligible: 00:00:08)

TemplateSets: DEFAULT
NodeMatchPolicy: EXACTNODE
Total Requested Tasks: 40

Req[0] TaskCount: 40 Partition: pbs
Dedicated Resources Per Task: PROCS: 1 MEM: 1280M
GMetric[energy_used] Current: 0.00 Min: 0.00 Max: 0.00 Avg: 0.00 Total: 0.00
NodeSet=FIRSTOF:FEATURE:[NONE]

Allocated Nodes:
[g-01-c0019:40]
Applied Nodeset: g-01-swil01

SystemID: Moab
SystemJID: 30518436

IWD: /home/projects/ku-cbd/people/shygop/C2_test
StartCount: 1
Execution Partition: pbs
Flags: RESTARTABLE
Attr: checkpoint
StartPriority: 124999
IterationJobRank: 0

shygop@g-12-10002 ~/projects/C2_test
$
```

- **Checkjob *job_id***

HOW DO I MONITOR MY JOBS?

- # • qstat

EXERCISE TIME - LET'S RUN A JOB

```
Canid1.R1.rastq.gz > Canid1.sam  
samtools view -b Canid1.sam >  
Canid1.bam  
samtools index Canid1.bam
```

```
#!/bin/bash  
#PBS -W group_list=ku_fa -A ku_fa  
#PBS -N canid1Map  
#PBS -e canid1.err  
#PBS -o canid1.log  
#PBS -l  
nodes=1:ppn=40,mem=50gb,walltime  
#PBS -d /home/projects/C2_test  
  
### Load modules  
module load bwa/0.7.15  
module load samtools/1.9  
module load htslib/1.9  
  
### Run your jobs  
bwa mem -t 40 refgenome/canFam31.fasta
```

OTHER WAYS TO LAUNCH JOBS: IN LINE PBS OPTIONS

```
#!/bin/bash
### Load modules
module load bwa/0.7.15
module load samtools/1.9
module load htslib/1.9

### Run your jobs
bwa mem -t 40 refgenome/canFam31.fasta Canid1.R1.fastq.gz >
Canid1.sam
samtools view -b Canid1.sam > Canid1.bam
samtools index Canid1.bam

qsub -W group_list=ku_fa -A ku_fa -N canid1Map -e canid1.err -o canid1.log -l
nodes=1:ppn=40,mem=50gb,walltime=1:00:00 -d /home/projects/C2_test mapCanid2.sh
```

OTHER WAYS TO LAUNCH JOBS: IN LINE PBS OPTIONS

```
echo "#! /bin/bash\n\n### Load modules\nmodule load bwa/0.7.15\nmodule load samtools/1.9\nmodule load htslib/1.9\n\n### Run your jobs\nbwa mem -t 40 refgenome/canFam31.fasta Canid1.R1.fastq.gz > Canid1.sam\nsamtools view -b Canid1.sam > Canid1.bam\nsamtools index Canid1.bam\n\" | qsub -W group_list=ku_fa -A ku_fa -N canid1Map -e canid1.err -o canid1.log -l\nnodes=1:ppn=40,mem=50gb,walltime=1:00:00 -d /home/projects/C2_test
```