

The acceptable face of integration development at UCL

Alexis Nelson
API & Application Integration Services
Information Services Division (ISD)
University College London

A single point of infinite temperature and density

A time of perfect order

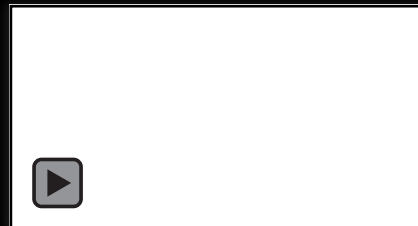
Limited options from Deliveroo



A single point of infinite temperature and density

A time of perfect order

Limited options from Deliveroo





particles

galaxies

fields

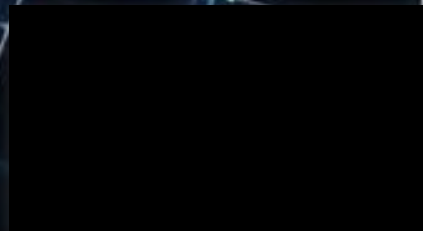
stars

planets

Chaos ensues

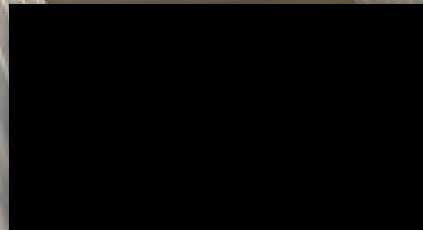
life

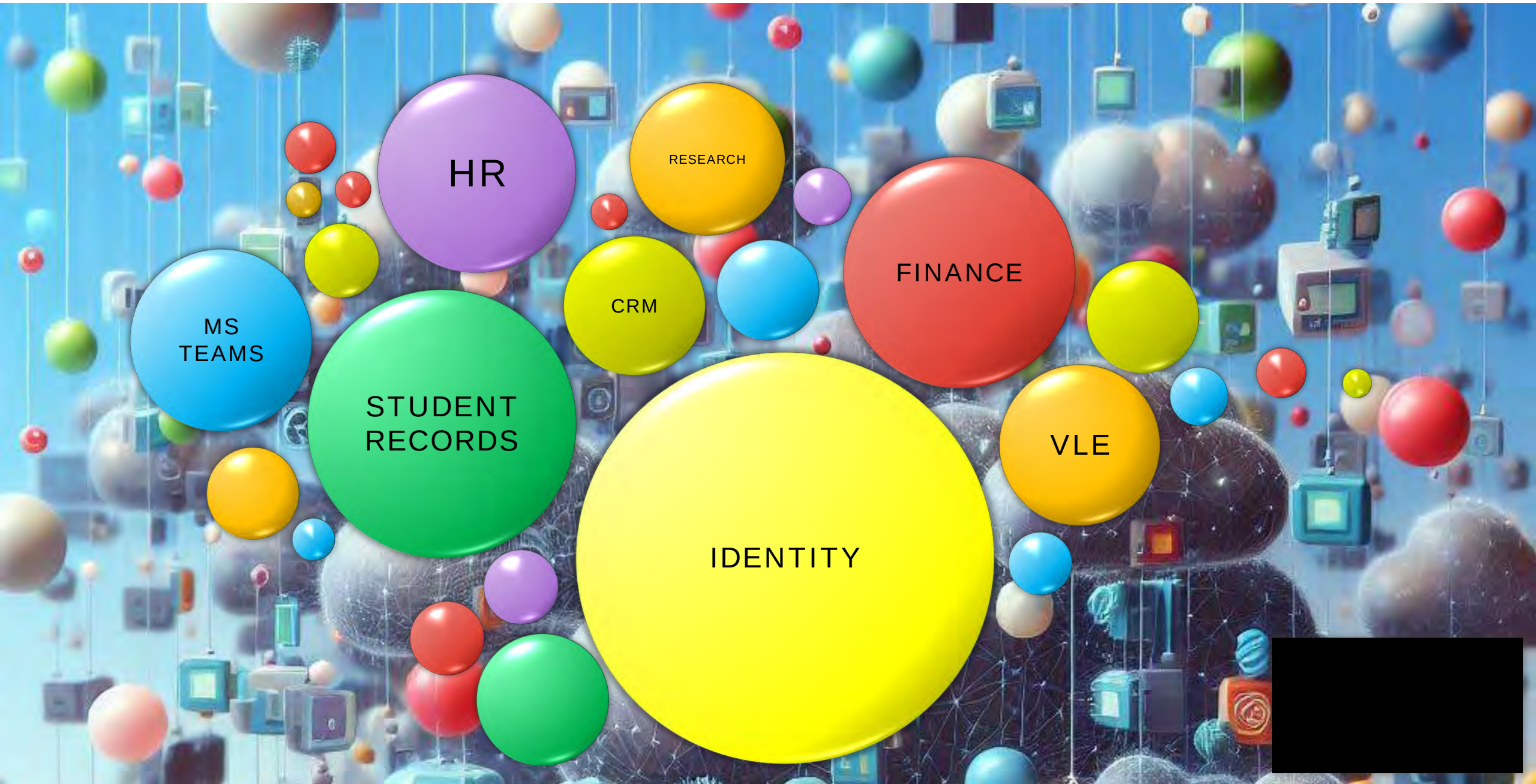
civilisation

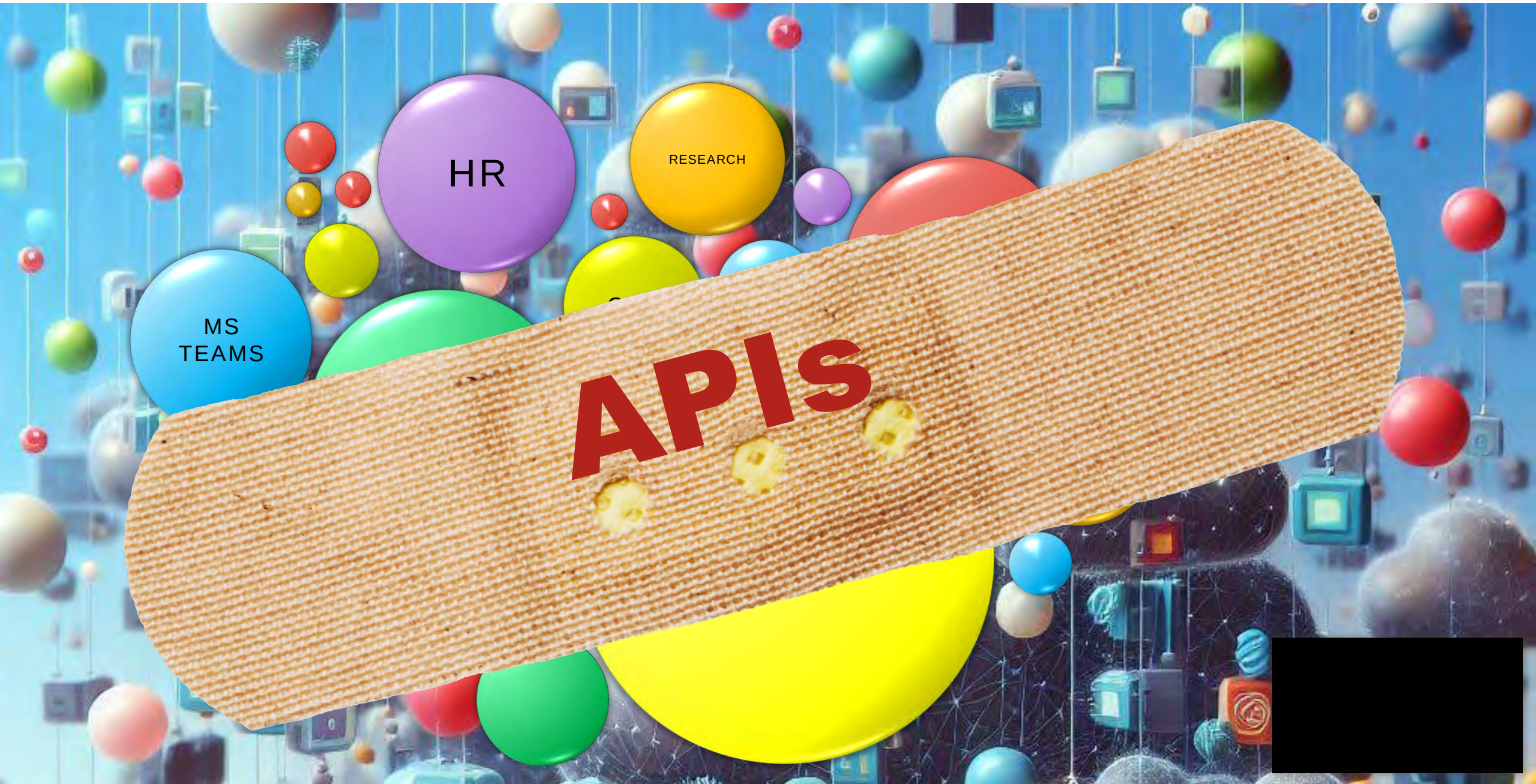


A low-angle photograph of a classical building facade, showing several tall, fluted columns supporting a pediment. The columns are made of light-colored stone or marble. The sky is a clear, bright blue.

ISD





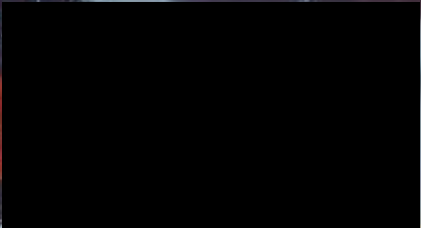


HR

RESEARCH

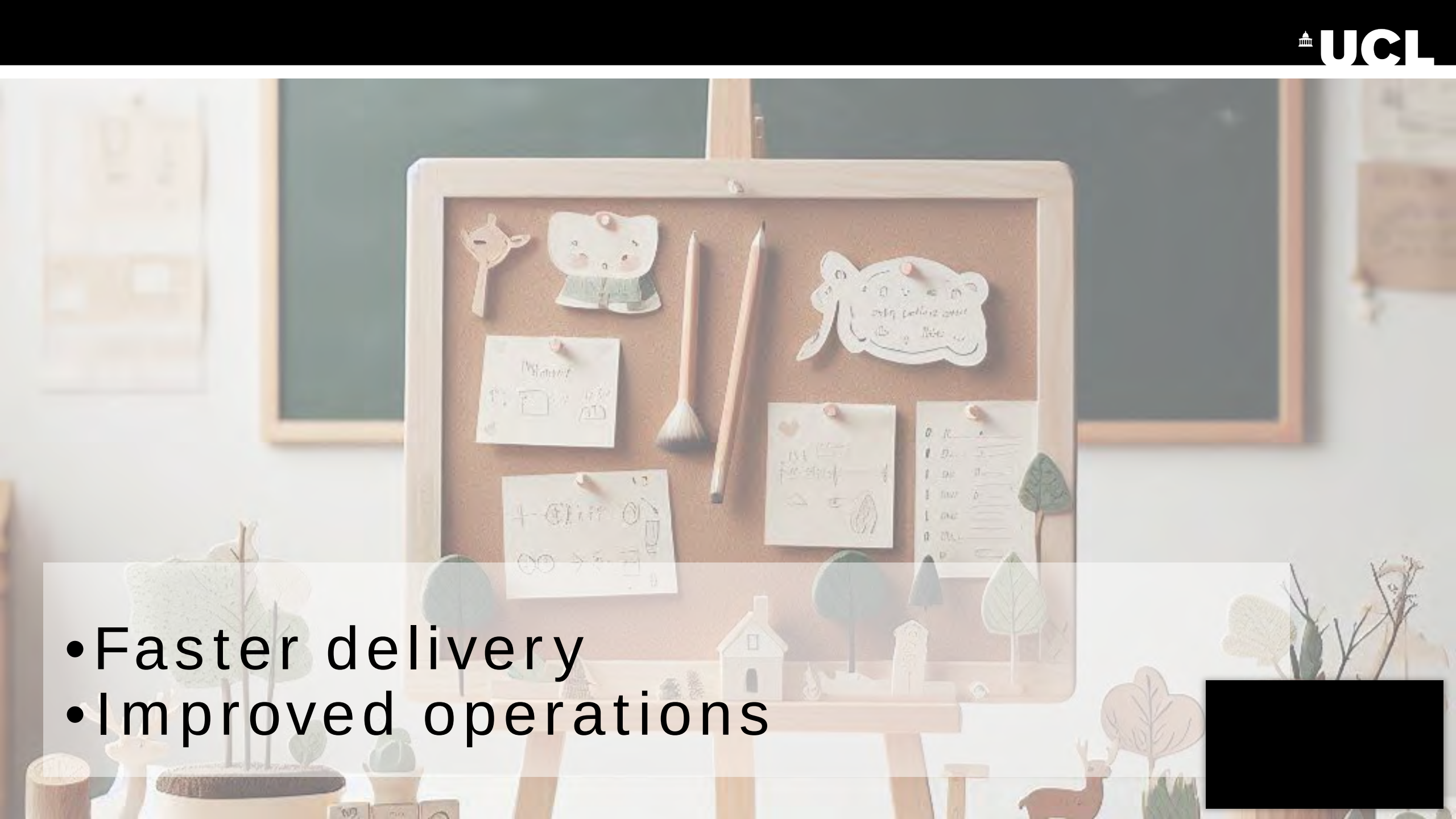
MS
TEAMS

APIs



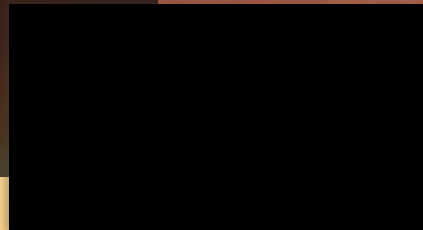


Back to the drawing board

- 
- Faster delivery
 - Improved operations

Laying the foundations

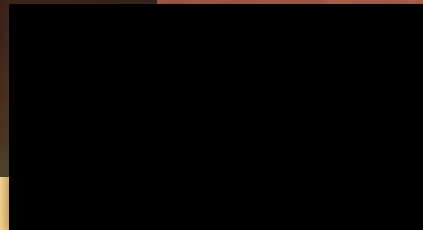
People
everyone owning their own interfaces



Laying the foundations

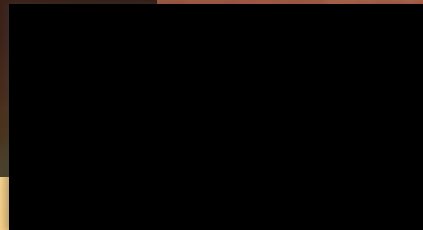
Process

simplifying and streamlining the path to delivery



Laying the foundations

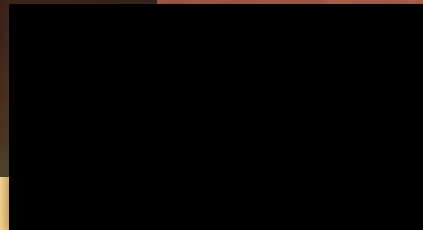
Data
governance to ensure consistency



Laying the foundations

Technology

throwing trendy buzzwords at the wall
until something sticks



The background of the slide is a detailed, high-angle illustration of a futuristic city. The city is composed of numerous tall, dark, rectangular skyscrapers with glowing windows and doors. The streets are filled with small, colorful lights and patterns, suggesting a complex, interconnected urban environment. In the center of the city, there are two large, bright green, oval-shaped islands. These islands are covered in a dense forest of small, dark trees. On each island, there are many small, dark figures of people, suggesting a large population. The islands are surrounded by a glowing blue energy field. The overall scene is a blend of advanced technology and nature, representing a sustainable, integrated urban environment.

introducing
Producer / Consumer TM
the new normal for integration conversations

Taking the high code



Industry standard software best practice
Transferable skills and personal development



Bridging capability gaps in teams
Preventing the inexorable onset of chaos



Enterprise Application & Service Integration toolkit

- Prevent entropy
- Enable adoption

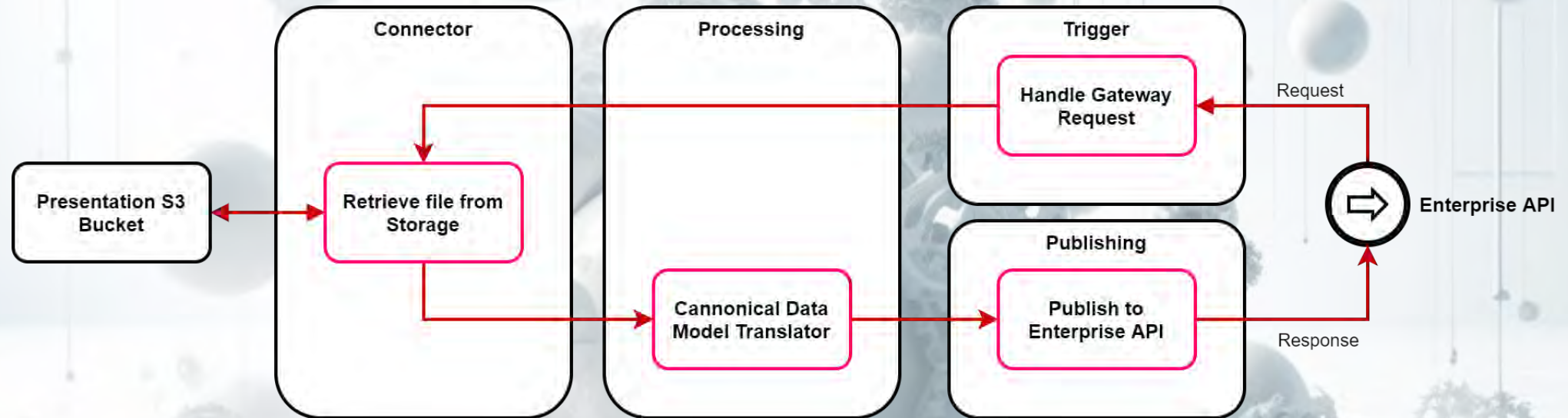
Enterprise Application & Service Integration toolkit

- Prevent entropy
- Enable adoption

EASIKit!



It begins with a conversation...



Molecules, atoms and cells, oh my!

» e python create

```
def handler(event, context={}):  
    req = event["easikit"]["request"]  
  
    if req["route_key"] == "GET /entity/{identifier}":  
        resp = sv.get_entity(req)  
  
    elif req["route_key"] == "POST /entity":  
        resp = sv.post_entity(req["body"])  
  
    else:  
        resp = er.RequestError(type=er.REQ_UNHANDLED_ROUTE, log=req)  
  
    return response.response(**resp, req=req)
```

Molecules, atoms and cells, oh my!

» e deploy create

```
{  
  "deploy_version": "0.1.0",  
  
  "config": {  
    "name": "easikit-ucisa-conference-demo-api-publisher",  
    "image_ecr": "123456789100.dkr.ecr.eu-west-2.amazonaws.com",  
    "image_repo": "easikit/ucisa-api-publisher:0.1.0",  
  
    "domain_name": "easikit.mesh.ucl.ac.uk",  
    "api_version": "v0.1",  
    "authoriser": "NONE",  
    "client_id": "NONE"  
  }  
}
```


Molecules, atoms and cells, oh my!

» e molecule atom s3

```
module "s3" {  
  source = "git@github.com:ucl-isd/easikit-terraform-s3.git//module"  
  
  # propagated from the terraform.tfvars.json config element  
  bucket_name      = local.config.bucket_name  
  
  # core variables, from variables.tf, always just here  
  name             = var.name  
  serv             = var.serv  
  env              = var.env  
  vrsn             = var.vrsn  
}
```


Molecules, atoms and cells, oh my!

» e docs snippets easiutils-awsutils-s3

```
def download(file_path)
# standard download to disk, extract the content to a variable
# the local download is deleted after the content is extracted
content = s3.download(file_path)

# to just download the file and get back the /tmp path it was downloaded to
download_path = s3.download(file_path, download_content=False)

# by default files are downloaded to /tmp/{file_path}
# to download somewhere (esp if downloading multiple files of the same name)
content_alt_path = s3.download(file_path, local_path="/tmp/alt-path")
```


Molecules, atoms and cells, oh my!

» easikit

**Python
function**

Business logic and flow

**Terraform
deployments**

Manage cloud services

**Python
libraries**

Standards/connectors

**Terraform
modules**

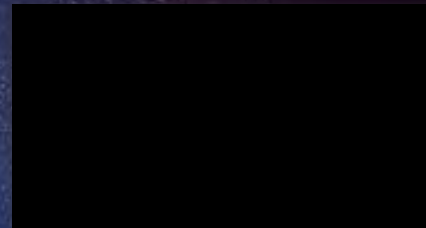
Additional services



The road more travelled

- **Launched summer 2022**
- **Multiple projects delivered**
- **8 cohorts trained**

Questions?



Post credits

<https://easikit.mesh.ucl.ac.uk/ucisa/v1/image>



Questions?

