AI4ME
AI for Personalised Media Experiences

Lancaster University

Agentic Object-based Media Delivery

the good , the bad and the ugly

Daniel King, Lancaster University *d.king@lancaster.ac.uk*
Co-authors: Rajiv Ramdhany (BBC), Edward Austin and Nicholas Race (Lancaster University)

October 5th, 2026
NIPA 2026 (IEEE ICNP Workshop)
Network Infrastructure and Protocols for AI Agents, Tempe, AZ, USA

Why a media-delivery talk at an agent's workshop?

- Several challenges at the workshop, including
  - **Intent interface**
  - **Discovery**
  - **Agent coordination across sessions**
  - **Scalability and security**
  - **Human (in-the-loop) Intervention**
- Object-based media (OBM) is a demanding, latency-sensitive workload where agents already make control decisions across media, compute and network domains.
- It is therefore a useful stress test for agent protocols, capability discovery and compute-aware networking.
- Everything here comes from a platform under active design, development and test deployments with the BBC: I will highlight what worked, what didn't, and what is still unsolved.

How to read the tags

GOOD	Patterns that held up: reuse of existing mechanisms, clean separations, things that work today
BAD	Problems we hit and have workarounds for, and provide an opportunity for industry standards
UGLY	Open, unsolved challenges.



Lancaster University

01

What is AI4ME

A BBC-led programme on AI for personalised media experiences, with Lancaster University (and others) on future media delivery, orchestration and standards

www.lancaster.ac.uk/scc/research/ai4me/

AI4ME: personalised media for mass markets

The ambition

- Content that adapts to each audience member: language, accessibility need, device, network
- Beyond recommendation and beyond bitrate ladders: the content itself changes

Why now

- Move from digital terrestrial TV towards full-IP delivery
- UK media green paper (2026) names personalisation as central to sustainable models

Inclusive by design

- Sight- and hearing-impaired audiences served by object variants, not a separate pipeline
- Accessibility features are protected first when resources are tight

Where Lancaster sits

- Delivery and orchestration: compute-aware steering, edge rendering, agentic control plane
- Open-source demonstrator for intent-driven edge provisioning
- Standards: AAIF, IETF, WC3: architecture, discovery, oversight, trust and security.

The distributed pipeline

- BBC production and partner systems
  - CDN, Cloud Compute, Network
  - User Experience
- Assembly can happen at any of these points, per viewer
- Every hop is a candidate for an agent, and a candidate for a failure

AI4ME research streams

Orchestration of OBM delivery

Automated resource allocation across device, edge and cloud; on-path compute for incremental assembly; RL-driven cache management

Adaptive content encoding

AI-driven object-level encoding assigns bitrates by perceptual importance, cutting power and distribution cost

Content generation

Generative models for personalised narratives; placing the generation itself is a compute-placement problem

Next-generation networks

6G and LEO impact on AI and compute placement; synchronised delivery across diverse network types

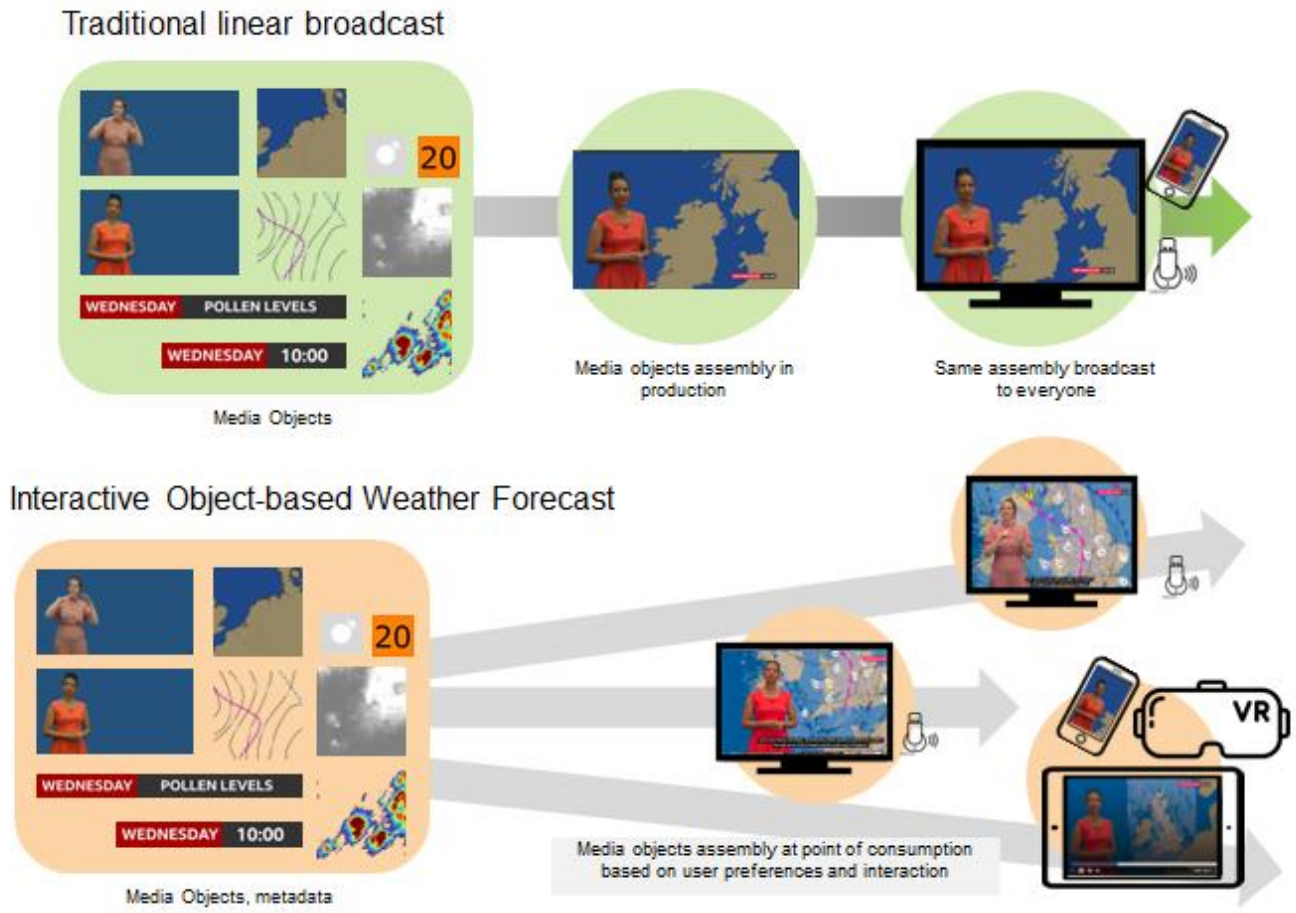
This talk is the first stream: delivering non-linear media at scale, and the agents that make the placement and steering decisions.

02

What is object-based media?

A programme is not a stream; it is a set of independent objects assembled per viewer, at the point of consumption or anywhere upstream

From one linear stream to a set of objects



- Media objects: audio, video, graphics, captions, 3D models, data streams, textures, shaders
- Metadata and rules describe how objects combine
- Assembly happens at the client, or at any point in the network
- Assembly can change as the user interacts, triggering re-rendering, re-composition, or inference

Image: BBC R&D / AI4ME

Use case: the interactive weather forecast

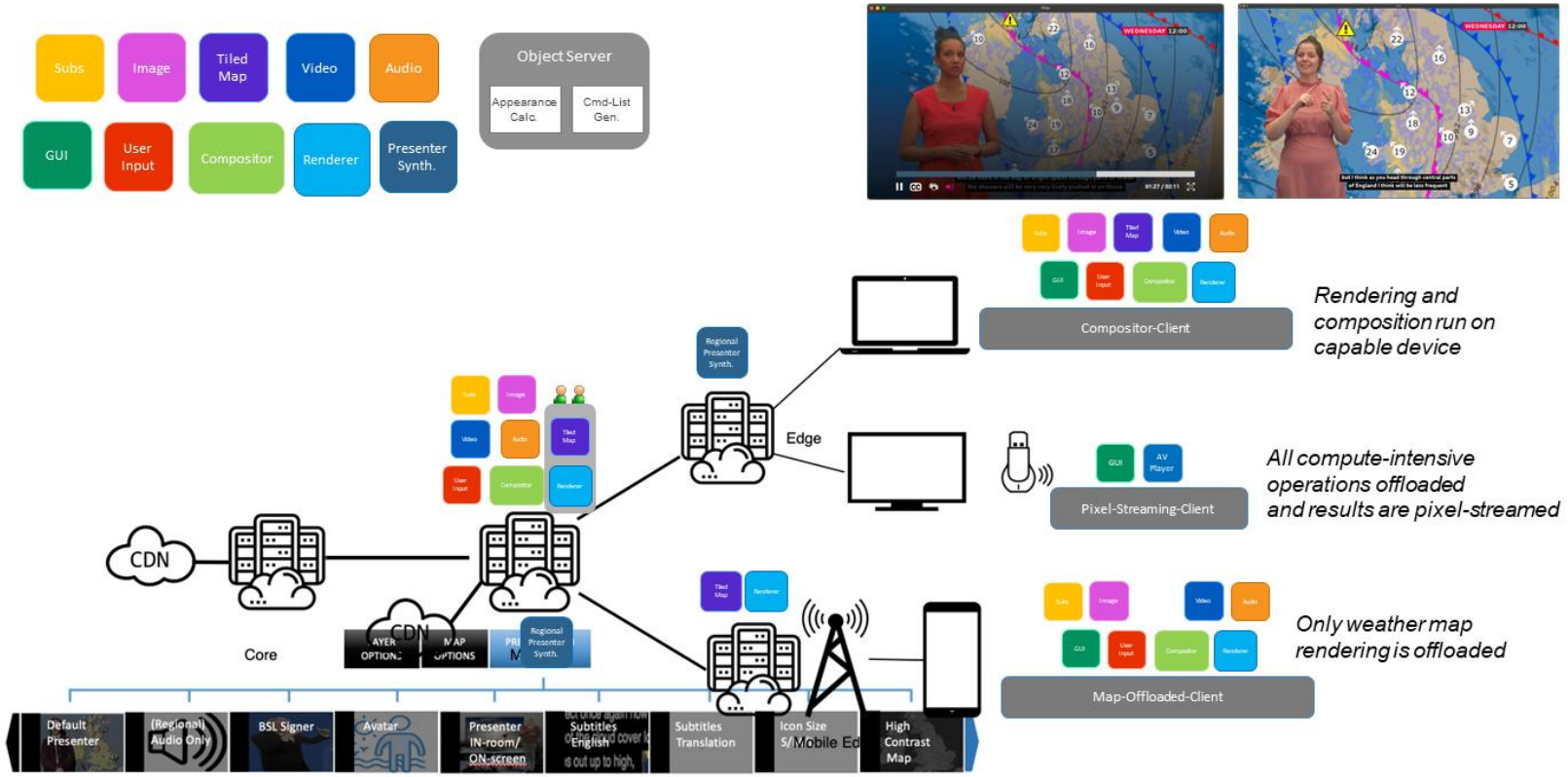


Image: AI4ME, IETF 122 HotRFC

- Same experience, three delivery shapes depending on device capability
- Rendering and composition on a capable device
- Compute-intensive operations offloaded and pixel-streamed to constrained devices
- Only the map rendering offloaded for mid-range devices
- The network, edge and core all take part, per session

BAD

Why OBM breaks the CDN delivery model

Combinatorial variants

Permutations grow with object count × user context. You cannot pre-render and cache every variant.

Workload moves to compute

Storage- and cache-centric delivery becomes compute-centric: each session can trigger a different pipeline, at a different place.

Split rendering is mandatory

Many devices cannot re-render or re-compose objects, so render offload to edge or cloud is needed for universal access.

Three dimensions now need joint control

(i) media composition logic (ii) compute placement (iii) network steering, in packet and optical layers

Existing systems handle each in isolation: CDN caching, DiffServ classes and TE pre-position content, not compute, and treat the network as blind to application compute demand.

Scale anchors

5–8
Mbps per HD (1080p30) session

15–25
Mbps per 4K30 session

~60 / ~1,200
4K / HD sessions per compute node

4
canonical workload classes

Workload class	Users	HD bandwidth (Mbps)	4K bandwidth (Mbps)	Aggregate (Gbps)
Narrative versioning	10,000	25,000–40,000	75,000–125,000	100–165
	1,000,000	2,500,000–4,000,000	7,500,000–12,500,000	10,000–16,500
Layered compositing	10,000	25,000–40,000	75,000–125,000	100–165
	1,000,000	2,500,000–4,000,000	7,500,000–12,500,000	10,000–16,500
Rendered objects	10,000	25,000–40,000	75,000–125,000	100–165
	1,000,000	2,500,000–4,000,000	7,500,000–12,500,000	10,000–16,500

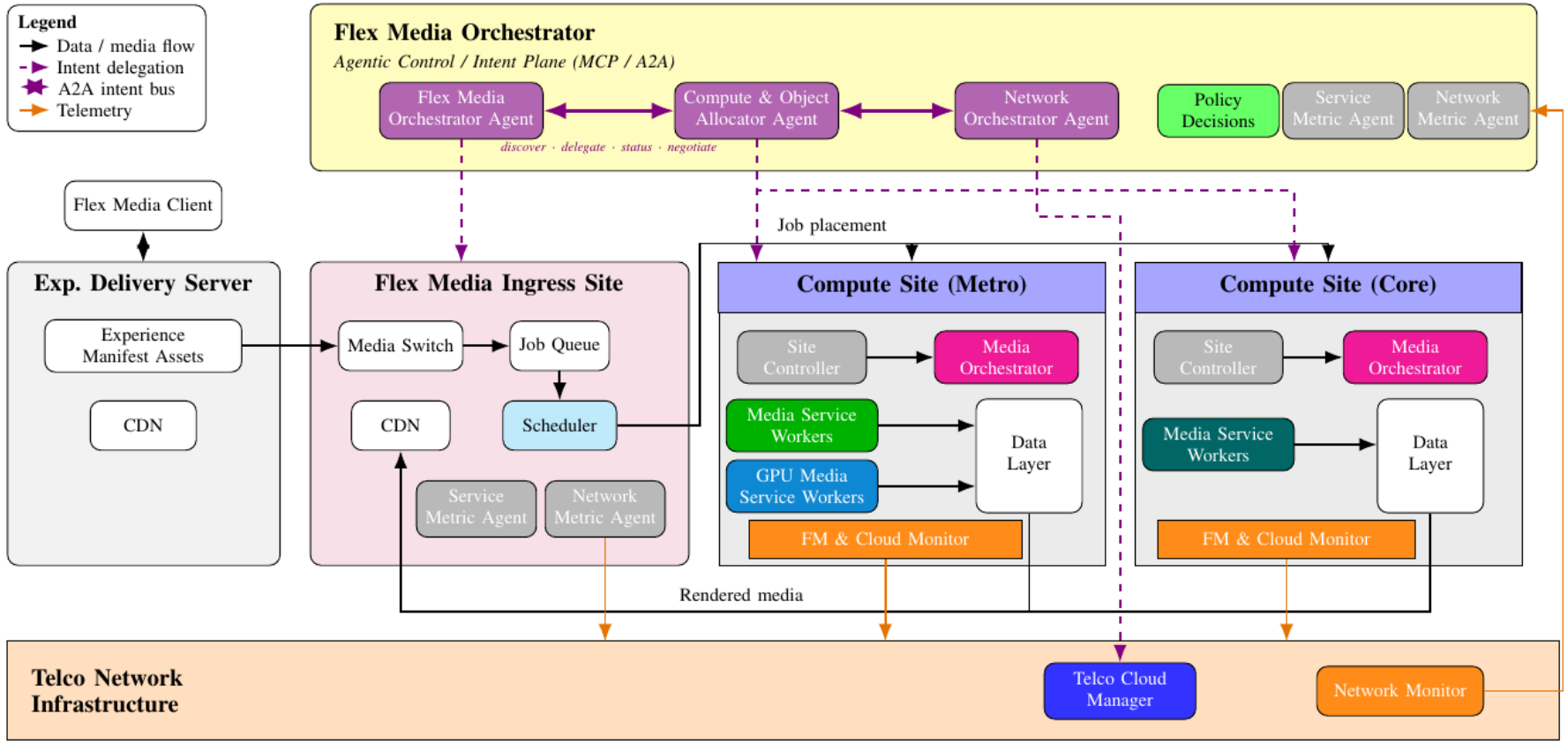
Workload classes from draft-rrk-object-based-media-usecase: narrative versioning, layered compositing, rendered objects, non-graphical objects. Figures are planning points, not capacity claims.

03

The OBM delivery architecture in AI4ME

A compute-aware Flexible Media platform: six planes, an agentic intent plane on top, mapped onto IETF CATS and ACTN

Reference architecture: six cooperating planes



Experience
Client, service, manifest server
Ingress
Media switch, job queue, scheduler
Compute
Metro and core sites, worker pools
Control
Flex Media Orchestrator: agentic intent plane
Infrastructure
Telco network, cloud manager
Observability
Metric agents and monitors, all planes

A Compute-Aware Architecture for Object-Based Media Delivery (IEEE CSCN 2026)
King, Ramdhany, Austin, Race

Functional entities and where they sit

Entity	Plane	Role
Flex Media Client / Experience Delivery Server	Experience	Requests experience; returns manifest and schema, fronted by CDN
Media Switch	Ingress	Resolves cache hits; forwards misses to the job queue
Scheduler	Ingress	Selects compute site(s) using telemetry and policy
Site Controller / Media Orchestrator	Compute	Accepts jobs; schedules CPU and GPU worker pools and storage within a site
Media Service Workers (CPU, GPU)	Compute	Composition, rendering and AI tasks
Flex Media Orchestrator Agent	Control	Experience intent, personalisation, session QoE policy; steers clients to ingress sites
Compute & Object Allocator Agent	Control	Compute-site lifecycle, capacity and placement
Network Orchestrator Agent	Control	Inter-site connectivity and transport path selection
Policy Decisions	Control	Cost, latency, energy and QoE trade-offs
Service / Network Metric Agents, Monitors	Observability	Aggregate service, compute and network telemetry; close the loop
Telco Cloud Manager / Network Monitor	Infrastructure	Underlay provisioning; transport-level metrics

End-to-end flow: four phases, nine steps

1 Manifest-first

(1.1) Client requests an experience and (1.2) receives manifest and schema from the Experience Delivery Server

2 Ingress placement

(2.1) Media switch resolves cache hits; misses become jobs with object IDs, frame windows and QoE targets; (2.2) scheduler picks sites on locality, RTT, job type, bandwidth & utilisation

3 Per-site orchestration

(3.1) Site controllers hand jobs to local media orchestrators, (3.2) which schedule worker pools, storage and local CDN

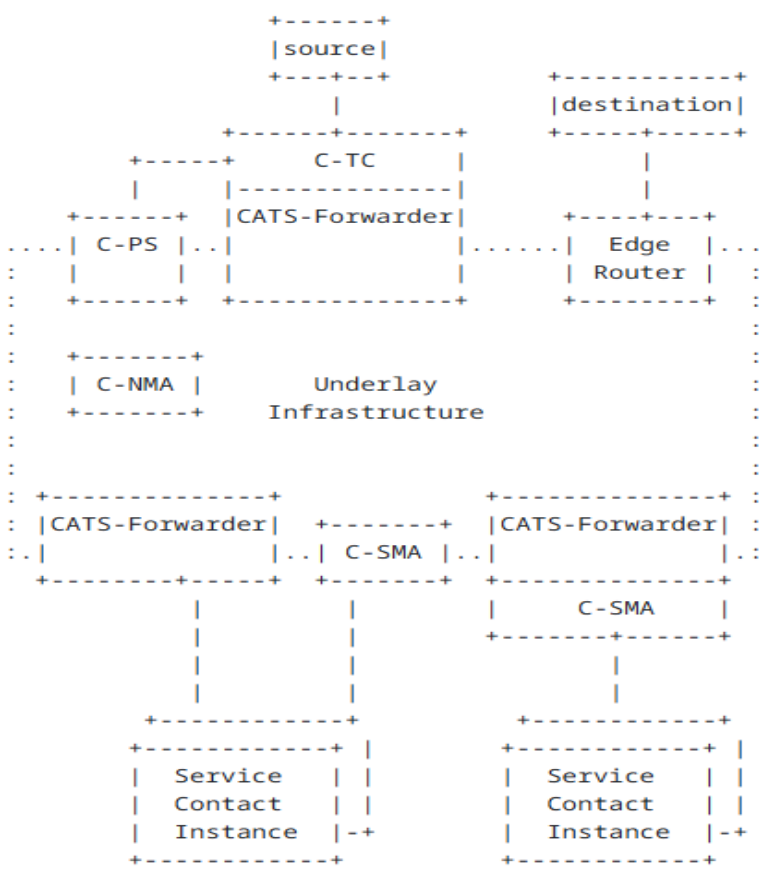
4 Global orchestration

(4.1) Composed output streamed in time for on-device composition; (4.2) telemetry closes the loop for policy, (4.3) scaling and inter-site connectivity adjusted as required

Interactive sessions resolve at ingress on a cache hit; rendering-heavy periods scale worker pools per site. Policy boundaries stay explicit so they can be tested and profiled for interoperability.

GOOD

Mapping onto IETF Computing-Aware Traffic Steering



CATS component	AI4ME entity
Path Selector (C-PS)	Ingress Scheduler + Network Orchestrator Agent
Service Metric Agent (C-SMA)	Service Metric Agents
Network Metric Agent (C-NMA)	Network Metric Agents
Traffic Classifier (C-TC) + Ingress Forwarder	Media Switch
Service contact instance	Site-local worker pool
CATS Service ID (CS-ID)	Composition stage

Objective was “to reuse, don't reinvent”

CATS supplies joint compute-network steering, ACTN the multi-domain control hierarchy, the OBM draft the media semantics.

CATS: draft-ietf-cats-framework, draft-ietf-cats-usecases-requirements, draft-ietf-cats-metric-definition
OBM use case: draft-rrk-object-based-media-usecase
Demonstrator: github.com/austine94/CNSM26

04

What agentic technologies are we using?

Entity discovery, cooperating agents, a policy function, telemetry agents, and two open protocols for the interfaces between them.

An agent, in this system

*A software **entity** that can perceive its environment, reason over goals, act on delegated tasks, and communicate with other agents. See draft-rrk-object-based-media-usecase*

Flex Media Orchestrator Agent

- Owns experience-level policy: personalisation, editorial rules
- Decomposes delivery goals into media-processing task templates
- Steers clients to ingress sites; holds session QoE objectives
- Configures the enforcement points

Compute & Object Allocator Agent

- Negotiates placement across distributed CPU, GPU and storage
- Tracks job state; reports resource availability
- Compute-site lifecycle, capacity and scaling
- Resolves resource conflicts

Network Orchestrator Agent

- Translates delivery requirements into path and steering decisions
- Reports latency, bandwidth and congestion state
- Coordinates with CATS-style service selection
- Inter-site connectivity

Agent Discovery: MCP & A2A: which protocol, for what?

Agent (entity) Discovery: end-point discovery, resource, capability, skill, task, et al.

MCP: agent to tool

- Standardises how an agent reaches tools, data sources and contextual state
- In AI4ME: access to telemetry stores and to enforcement points (media switch, scheduler, site controllers)

A2A: agent to agent

- Peer collaboration between opaque agents: capability advertisement (Agent Cards), task delegation, lifecycle, async status
- In AI4ME: the inter-agent negotiation itself

Structured intents, not resource commands

Objective + constraints + scope: “place render stage” “enforce affinity” “cap latency” “limit cost growth”

Receiver acknowledges a scoped request, or returns a constrained counterproposal. Asynchronous status reports placement, steering and failure state. The originator records the converged action for audit and policy feedback.

The explicit interface separates policy logic from vendor-specific control implementations, so placement and steering decisions are auditable.

GOOD

Design choice: agents off the per-request path

Meta-level: the agents
Flex Media Orchestrator · Compute & Object Allocator · Network Orchestrator · Policy Decisions abd configure enforcement points, steer clients to ingress sites, set policy weights

Per-request path: no agent in the loop
Client · Media Switch (cache hit / miss) · Scheduler · Site Controller · Media Orchestrator · Workers are deterministic, local, fast

- LLM latency and non-determinism are kept out of the session-critical path by construction
- Agents are measured on their own timescale: intent acceptance latency, intent-to-action convergence time, policy conflict rate, intent success ratio

05

How agents are used in AI4ME

Personalisation, composition placement, resource optimisation, multi-stage coordination, QoE adaptation and capability negotiation

Six practical OBM functions for agents

Personalisation

Choose language, accessibility, camera angle, storyline and object variants per user

Dynamic composition

Decide whether content is assembled on-device, at the edge, or upstream

Resource optimisation

Place rendering and composition tasks on the most suitable compute nodes

Multi-stage coordination

Steer a session through compilation, rendering, caching and final delivery

QoE adaptation

Degrade non-essential enhancements first; preserve core content and accessibility features

Capability negotiation

Discover what peer systems support; negotiate resources and tasks across domains

One policy, five weights: how placement is scored

$$S(p, s) = w_q Q - w_l L - w_c C - w_e E - w_r R, \quad \Sigma w_i = 1$$

Q

predicted QoE gain

object asynchrony, adaptation smoothness, quality-switch behaviour

L

latency penalty

end-to-end, from network telemetry

C

infrastructure cost

per-session processing cost, cache-hit ratio

E

energy impact

energy per delivered minute

R

reconfiguration risk

affinity violations and re-steering churn over the hold interval

Path p and service contact s are chosen by maximising S. Operators prioritise cost, energy or affinity by moving the weights, not by adding control machinery. Constrained transport profiles (deterministic latency class, hard-isolation path class) enter as eligibility filters on (p, s) and through R.

Telemetry that feeds the loop: six metric groups

Group	Key metrics	What it tells the agents
Delivery	Startup delay, rebuffer ratio, segment completion, end-to-end latency	Session responsiveness and playback continuity
QoE	Object asynchrony and tolerance, adaptation smoothness, quality-switch magnitude and frequency	Perceptual quality; feeds Q
Compute	CPU/GPU utilisation, queue depth, stage execution time, scheduler decision latency	Per-stage efficiency and headroom
Network	Throughput, loss, RTT, jitter, congestion indicators	Path stability for interactive delivery; feeds L
Cost / Energy	Cost per session, cache-hit ratio, energy per delivered minute	Economic and environmental efficiency; feeds C and E
Intent plane	Intent acceptance latency, intent-to-action convergence time, policy conflict rate, intent success ratio	Control-loop responsiveness and policy effectiveness

Instrumentation: client playback telemetry, ingress and compute service metrics, network probes, all with synchronised timestamps so causality can be traced across pipeline stages. Hardware classes (CPU, GPU, NPU) are distinguished so placement policies can be compared fairly.

BAD

Closing the loop is harder than opening it

Oscillatory re-steering

Raw, bursty telemetry from heterogeneous sites destabilises the control loops that consume it. Today: dampening and thresholded updates.

No normalisation guidance

Compute and network metrics from different vendors and tiers cannot be compared without a common profile. This is a standards gap, not a code gap.

Still characterising

Scheduler stability under bursty demand; signalling overhead for compute-state dissemination; stability of RL-assisted cache policies.

Feasibility evidence: open-source demonstrator

Implements the three-layer agent architecture and provisions edge compute for personalised live media sessions.

Public code and reproducible experiments (Austin et al., CNSM 2026, github.com/austine94/CNSM26).

Planned evaluation: does compute-aware steering deliver measurable QoE, cost and energy gains over compute-agnostic baselines? Those gains are what justify the added control-plane complexity.

06

Deep dive: agent discovery for OBM

How does the BBC's orchestrator find render capacity in a telco cloud it has never met, verify it, and reconnect to it mid-session?

The agent discovery problem, in OBM terms

Agent (entity) Discovery: end-point discovery, resource, capability, skill, task, et al.

BBC Flex Media Orchestrator Agent

Discovering entity. Needs a GPU effect-render stage for a live session, near a population of viewers.

Telco Cloud — Compute & Object Allocator

Discovered entity. Advertises capability (gpu-render, object-composition), locality, protocol (A2A/HTTPS) and a trust anchor.

Third-party CDN render pool

Alternative discovered entity. Different operator, different issuer, different policy system.

Telemetry stores, ingress sites

Resources the orchestrator must find and read (static capability + live state) to decide placement.

The gap: the BBC's and the telco's systems have never met. How does the orchestrator find an allocator at a known organisation, learn what it can do and how to talk to it, trust the answer, and reconnect to the same instance for the life of a session?

Four discovery requirements, mapped to OBM

01 Capability-oriented

Who can do this?

Which agent or site can run a GPU effect-render stage, an object-composition stage, a packaging stage, speaking A2A over HTTPS?

02 Resource-oriented

Where are suitable resources?

Sites with GPU class X, storage headroom, current queue depth and utilisation; a pointer to live telemetry rather than a stale snapshot.

03 Administrative scope

How far can discovery reach?

BBC → telco cloud → third-party CDN. Cross-organisation by construction; locality and jurisdiction filters (UK-only rights windows).

04 Operational

What is running, and who is responsible?

Which agent authorised which placement, for editorial and rights compliance, troubleshooting and audit.

Capability-oriented discovery: the query

Intent

“Place the effect-render stage for tonight's live session somewhere with GPU headroom and sub-20 ms RTT to the Manchester ingress site.”

```
Discovery query (MVD)

 Entity_Type:    Agent | Service
 Capability:     gpu-render,
                 object-composition
 Target_Org:     telco.example
 Protocol:       A2A / HTTPS
 Locality:       UK, metro=MAN
 Policy_Filter:  rights=UK-only
 Dynamic_Hint:   gpu_headroom>=20%
```

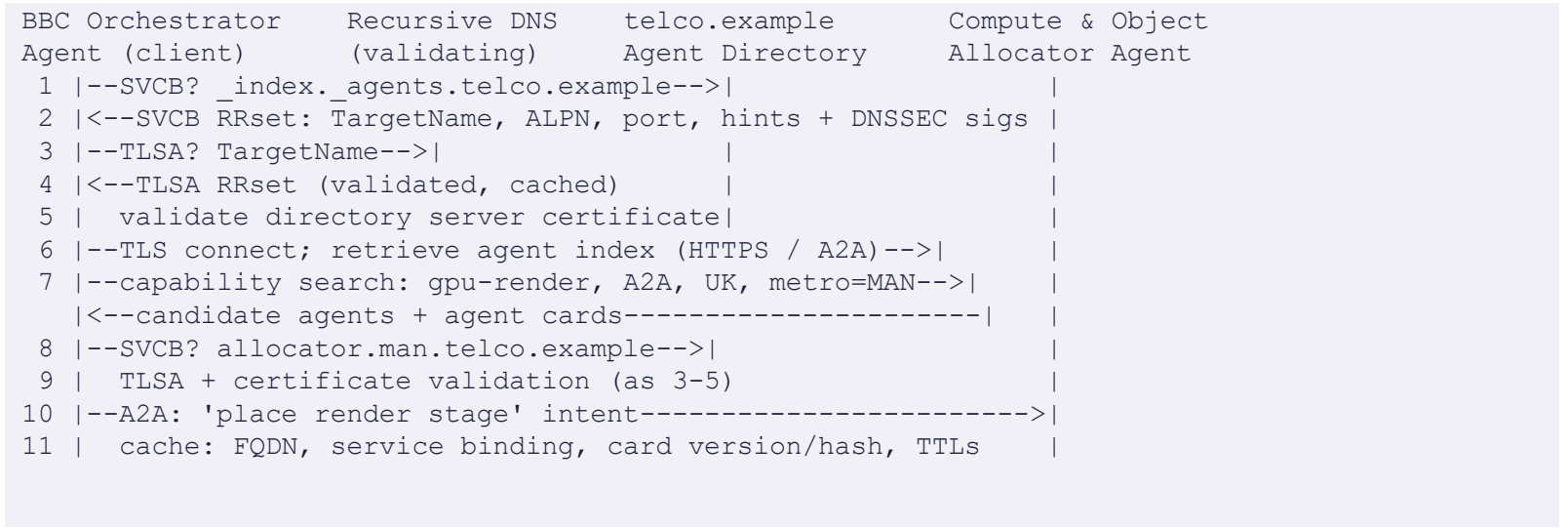
Returned (MDI)

Candidate agents with: FQDN, service binding (protocol, endpoint, port), capability tags, responsible organisation, trust indicators (DNSSEC/TLSA, card hash), pointer to the full agent card and to live state.

Selection and negotiation then happen over A2A, outside discovery.

GOOD

Cold start: known organisation, unknown capability

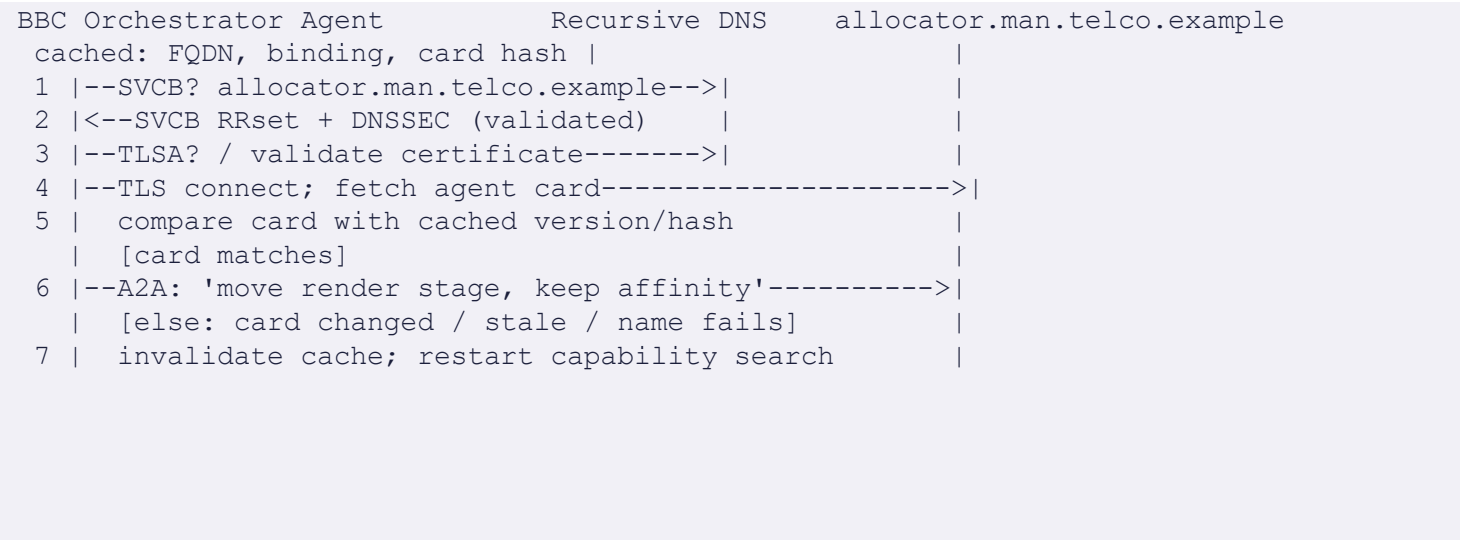


- DNS and mDNS are the bootstrap, not the catalogue
- The directory is pluggable: organisational index, ARD, or another service
- Trust comes from DNSSEC, TLSA and the card hash, before any A2A exchange
- Reuses deployed, organisationally devolved infrastructure

Based on DNS-AID (draft-mozleywilliams-dnsop-dnsaid) Figure 5, extended with DAWN capability search (6–7) and result caching (11).

GOOD BAD

Warm start: re-steering a live session



- No directory needed on a warm start: resolve, verify the card, go
- Trust in the metadata is not trust in the live entity; the card is stable, the site state is not
- Fallback makes discovery a loop: card changed, name failed, or TTL expired means back to the index
- Ties to affinity: re-selection between sessions is cheap, re-steering an active session is not

Based on DNS-AID Figure 2 (known agent discovery), extended with the card freshness check (5) and the fallback to a fresh capability search (7).

UGLY

Resource discovery: static metadata, live state

Static: belongs in discovery

- Hardware class (CPU, GPU, NPU), memory, storage tier
- Locality, jurisdiction, responsible organisation
- Supported stages and protocols, agent card version

Dynamic: a pointer, not a value

- Utilisation, queue depth, GPU headroom, power
- Changes in seconds; too volatile for the discovery layer
- Fetched via the pointer: the CATS metric agents, by another name

The ugly part: what discovery exposes

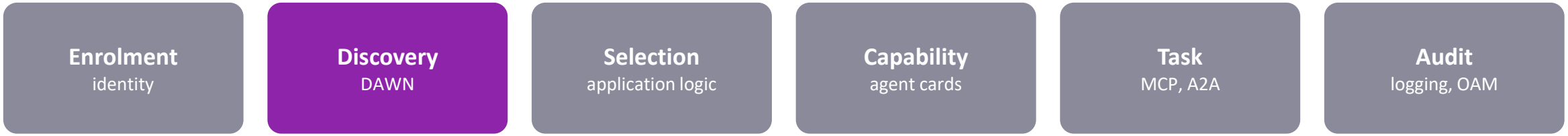
Compute availability, cache state, object metadata and session-derived policy inputs are all operationally sensitive. Across organisational boundaries they describe your capacity, your audience and your editorial decisions.

Needed: strict access control, scoped telemetry visibility, local policy interpretation where possible, and integrity protection on control and metric channels. Security and privacy remain under-specified in current OBM drafts.

Information model, and where discovery stops

MVD — minimum discovery data (the query)	OBM value
Entity type sought	Agent / service
Capability required	gpu-render, object-composition
Target	telco.example, or a named allocator
Protocol constraints	A2A over HTTPS
Locality / jurisdiction / policy	UK, metro=MAN, rights=UK-only

MDI — minimum discoverable information (the answer)	OBM value
Name (FQDN)	allocator.man.telco.example
Entity type, capability tags	agent; gpu-render, composition
Service binding	A2A, https, :443
Responsible organisation	telco.example
Trust indicators	DNSSEC, TLSA, card hash
Pointer to rich info and live state	agent card; metric agent endpoint



Discovery answers one question: "find me an entity to interact with", with just enough verified information to decide whether and how to connect. Enrolment, identity, selection, task execution and audit are someone else's problem, and that is deliberate.

IETF Discovery of Agents With Names (dawn)

What it is

- Discovery of Agents With Names, proposed IETF WG, Internet Area
- Chairs Adrian Farrel, Wes Hardaker; AD Éric Vyncke
- BoF at IETF 126 (Vienna, July 2026); charter under IESG review, on the 8 Oct 2026 telechat
- Reuses DNS/mDNS, DNSSEC and TLSA rather than inventing a registry

Scope, deliberately narrow

- In: discovering public properties of agents, workloads and named entities, locally, within an organisation, and across the Internet between trusted parties
- In: a minimum public discovery information model, an architecture, a protocol
- Out: capability catalogues, post-discovery protocols, Internet-wide search, identity and trust, semantic matchmaking

Deliverables and timeline

- Terminology (informational, optional, mid-2027)
- Use cases and applicability (informational)
- Discovery architecture (informational, early 2028)
- Protocol specification (standards track, to IESG December 2027)

Why it matters for OBM

- The cold-start and warm-start flows you just saw are the DNS-AID pattern DAWN builds on
- OBM cross-domain placement is one of the use-case inputs
- The out-of-scope list is exactly our lifecycle boundary: DAWN finds the entity (task, compute, CDN); MCP/A2A, CATS and the trust work take over from there

datatracker.ietf.org/group/dawn/about · github.com/danielkinguk/discovery · draft-jimenez-dawn-discovery-landscape · draft-mozleywilliams-dnsop-dnsaid

The good, the bad and the ugly

The good

- Intent abstraction separates policy from vendor control, and makes decisions auditable
- Agents sit off the per-request path; sessions never wait on an LLM
- Clean instantiation of CATS and ACTN; DNS, DNSSEC and TLSA reused for discovery bootstrap
- MCP and A2A exist under open governance, with agent cards as the capability contract
- Split rendering reaches every device; a working open-source demonstrator

The bad

- Raw telemetry destabilises control loops; dampening is a workaround, not a design
- CATS assumes one instance per request; on-path chaining is out of scope
- No metric normalisation or deployment-profile standards; every site is a special case
- Signalling overhead for compute-state dissemination; scheduler stability under bursts
- Cached discovery metadata is stable; the state behind it is not

The ugly

- No agreed substrate for identity, delegation, least privilege and audit across domains
- Oversight and human-in-loop control for concurrent agents propose competing actions; no conflict-resolution or authority model
- Discovery and telemetry leak capacity, audience and editorial state across organisations
- Agent protocol churn: the interface contract is a moving target
- Security and privacy under-specified in every OBM draft, including ours

Five asks

1. **Align OBM metadata with CATS service selection**
   Composition stages and service-contact abstractions mean the same thing in every deployment

2. **A minimal intent profile over A2A / MCP**
   Required fields for capability, objective, constraints, scope, acknowledgement, counterproposal, status and error

3. **Metric normalisation guidance**
   Common semantics plus update-rate controls (dampening, thresholds, aggregation windows) so loops stay stable

4. **Deployment profiles for control models**
   Centralised, distributed and hybrid; expected behaviour for affinity hold-time and re-steering boundaries

5. **Oversight (see next slide on ICON)**
   Guardrails for network operation actions, telemetry visibility and OBM monitoring across domains

IETF ICON: guardrails for agents that touch the network

What it is

- Intervention, Control and Observability of Network management agents
- Discussed in ETF AINETOPS use-cases work: AI for network operations across single- and multi-layer IP/optical networks
- Rokui (Ciena), King (Lancaster), Li and Wu (Huawei)
- Continuous visibility of agent behaviour; operator intervention; controls that stop unanticipated actions reaching the network

Use cases, all with guardrails

- Network active assurance with AI-agent support
- Anomaly detection
- AI-driven policy enforcement and compliance auditing
- Agentic resilience testing
- Human on the loop: operators submit high-level intents; agents run the workflow inside bounded blast radius

Why it matters for OBM

- Our Network Orchestrator Agent is an ICON-class agent from the telco's point of view
- When it turns “cap latency” into path and steering changes on someone else's network, that operator needs to observe, intervene and address it
- This is the authority-boundary problem from the ugly column; ICON is where its requirements are being written

Open questions we are asking

- Which agent actions need pre-approval, which post-hoc audit, which are forbidden?
- Can a guardrail be expressed in the same intent vocabulary as the request it bounds?
- Who owns the kill switch when the agent is the BBC's and the network is the telco's?

draft-king-rokui-ainetops-usecases · github.com/billwuqin/ICON-Use-Case · github.com/IETF-OPS-AD/AINETOPS

AI4ME
AI for Personalised Media Experiences

Lancaster University

In Summary

- OBM at scale needs joint control of composition, placement and steering; agents give us the intent plane to do it, agents are kept off the session path.
- Entity discovery can reuse DNS, DNSSEC and TLSA today; what remains ugly is trust, conflict and leakage across organisations.
- Oversight for agent network operations, telemetry and human-in-loop control, are unanswered for now, but IETF ICON looks promising

IETF Drafts: draft-rrk-object-based-media-usecase · draft-ietf-cats-framework · draft-kiliram-agent-trust-auth-framework · DAWN requirements
IETF ICON Side Meeting https://github.com/billwuqin/Agent-Observability-Intervention-Control
Code: github.com/austine94/danielkinguk
Code: github.com/austine94/CNSM26

Thank you, questions and follow-up welcome via
d.king@lancaster.ac.uk