CONTENT-CENTRIC COMMUNICATION

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

"Information-centric networking (ICN) is an approach to evolve the Internet infrastructure away from a host-centric paradigm, based on perpetual connectivity and the end-to-end principle, to a network architecture in which the focal point is identified information (or content or data). In this paradigm, connectivity may well be intermittent, end-host and in-network storage can be capitalized upon transparently, as bits in the network and on data storage devices have exactly the same value. Mobility and multi access are the norm and anycast, multicast, and broadcast are natively supported. Data becomes independent from location, application, storage, and means of transportation, enabling in-network caching and replication. The expected benefits are improved efficiency, better scalability with respect to information/bandwidth demand and better robustness in challenging communication scenarios."
(Wikipedia)

BASIC REQUIREMENTS

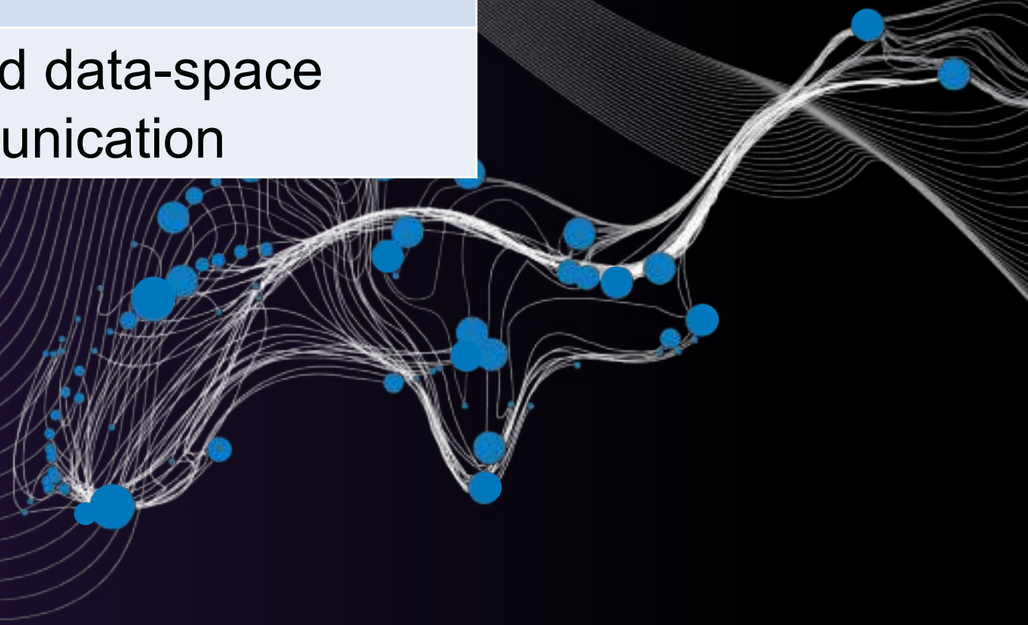
- Look up objects (notably files) based on their content (“contains keyword X”) instead of location (“F@smb://domain/path”)
- Needs agreement on what content stands for (this is not trivial)
- Need facilities for finding and exploring content
- Need for massive scalability: size, geographically, across administrative boudaries

RELATED IN MODERN DATA SCIENCE: FAIRness

- Findable
- Accessible
- Interoperable
- Reusable

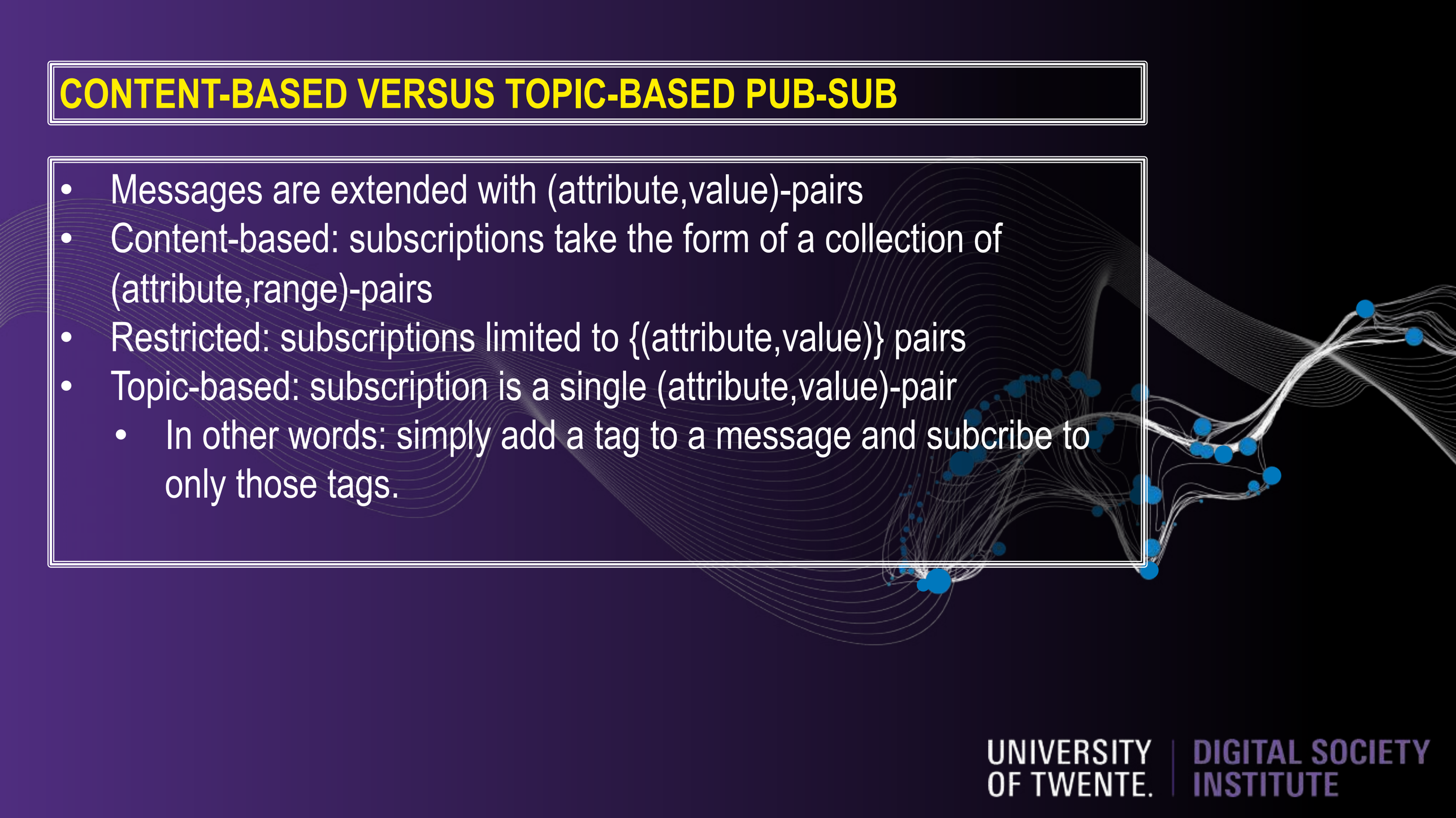
TIME-SPACE COUPLING OF COMMUNICATING PARTIES

	Temporally coupled	Temporally decoupled
Referentially coupled	Point-to-point communication	Mailbox communication
Referentially decoupled	Event-based communication	Shared data-space communication

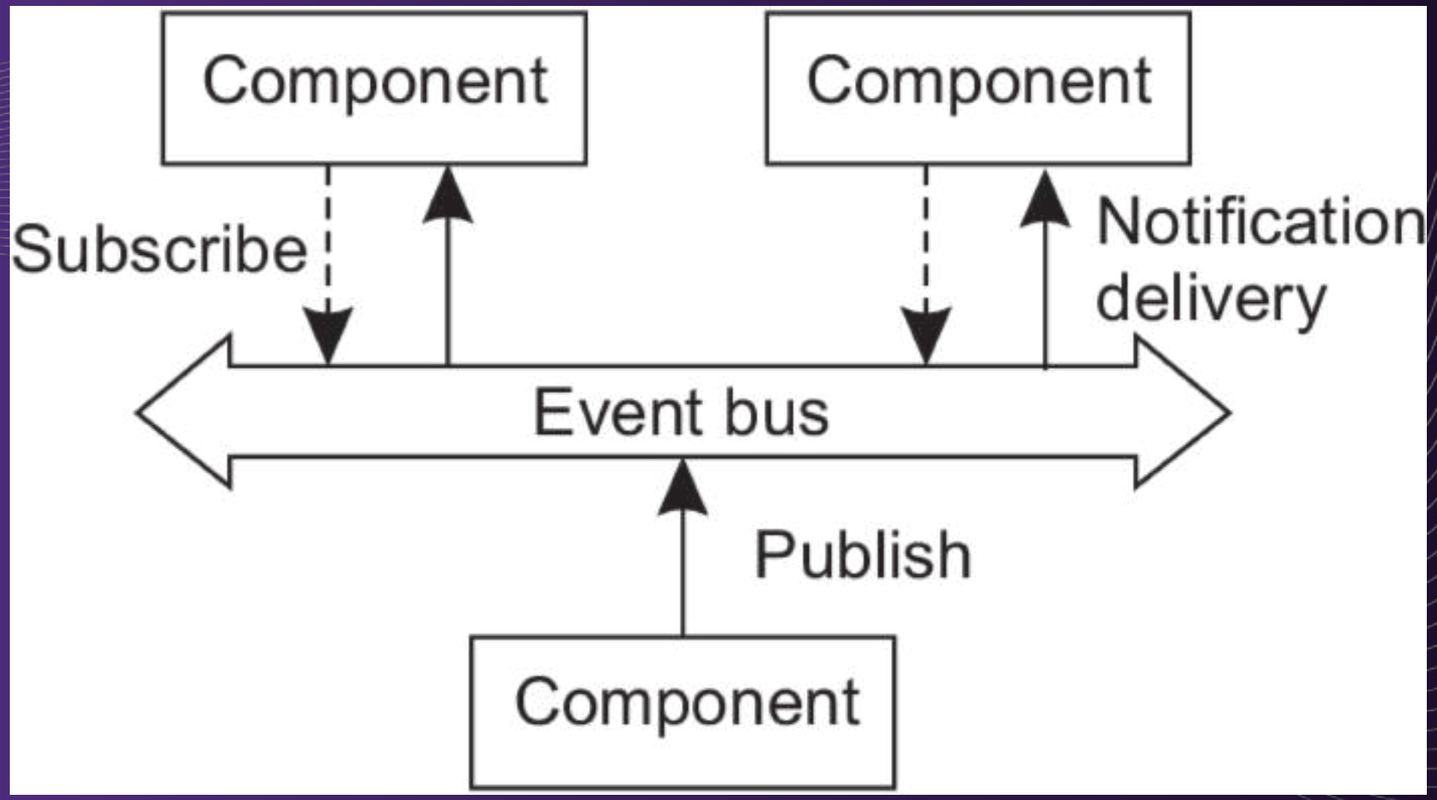


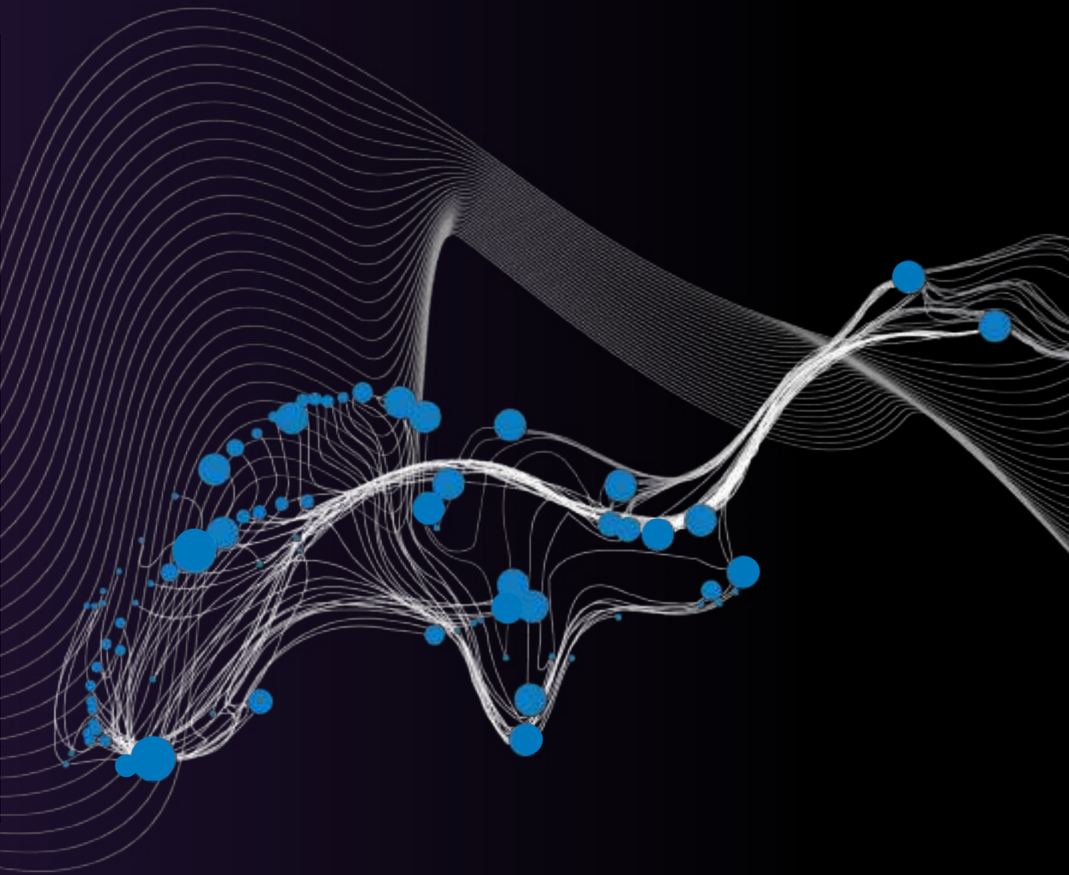
CONTENT-BASED VERSUS TOPIC-BASED PUB-SUB

- Messages are extended with (attribute,value)-pairs
- Content-based: subscriptions take the form of a collection of (attribute,range)-pairs
- Restricted: subscriptions limited to {(attribute,value)} pairs
- Topic-based: subscription is a single (attribute,value)-pair
  - In other words: simply add a tag to a message and subcribe to only those tags.



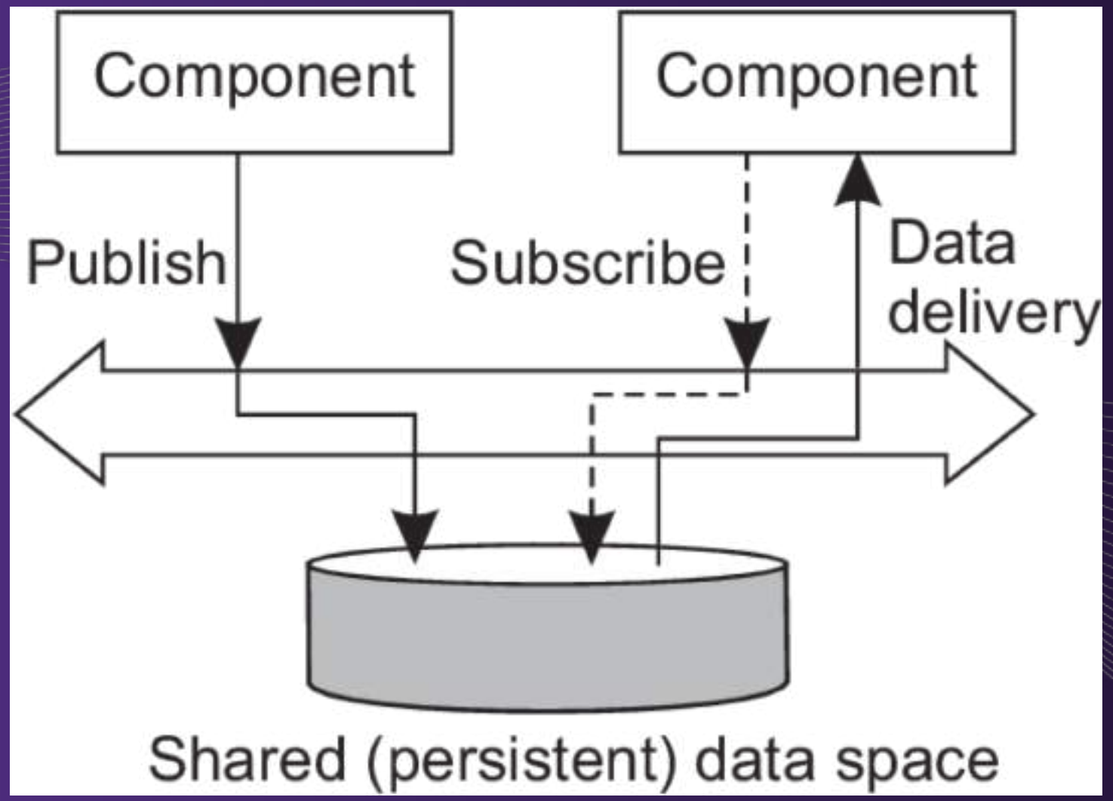
EVENT-BASED COMMUNICATION

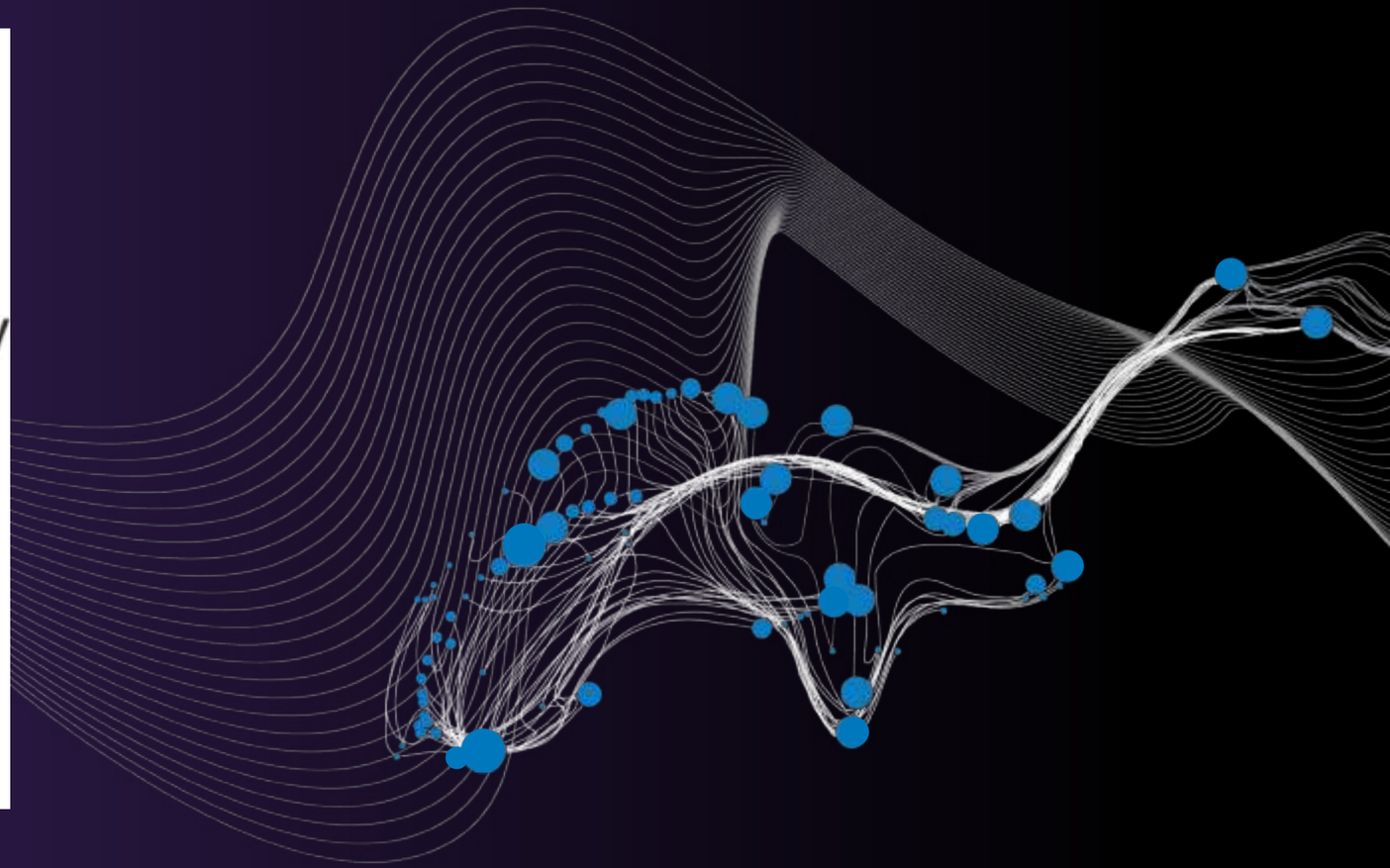




IMPORTANT: communicating parties need to be both online

SHARED DATA-SPACE COMMUNICATION





IMPORTANT: data to communicate can be stored

EXAMPLE: LINDA TUPLE SPACE

OPERATIONS:

- in(t): remove a tuple that matches the template t
- rd(t): obtain a copy of a tuple that matches the template t
- out(t): add the tuple t to the tuple space

BOB:

```
1 blog = linda.universe._rd(("MicroBlog",linda.TupleSpace))[1]
2
3 blog._out(("bob","distsys","I am studying chap 2"))
4 blog._out(("bob","distsys","The linda example's pretty simple"))
5 blog._out(("bob","gtcn","Cool book!"))
```

ALICE:

```
1 blog = linda.universe._rd(("MicroBlog",linda.TupleSpace))[1]
2
3 blog._out(("alice","gtcn","This graph theory stuff is not easy"))
4 blog._out(("alice","distsys","I like systems more than graphs"))
```

CHUCK:

```
1 blog = linda.universe._rd(("MicroBlog",linda.TupleSpace))[1]
2
3 t1 = blog._rd(("bob","distsys",str))
4 t2 = blog._rd(("alice","gtcn",str))
5 t3 = blog._rd(("bob","gtcn",str))
```

PROBLEMS AHEAD

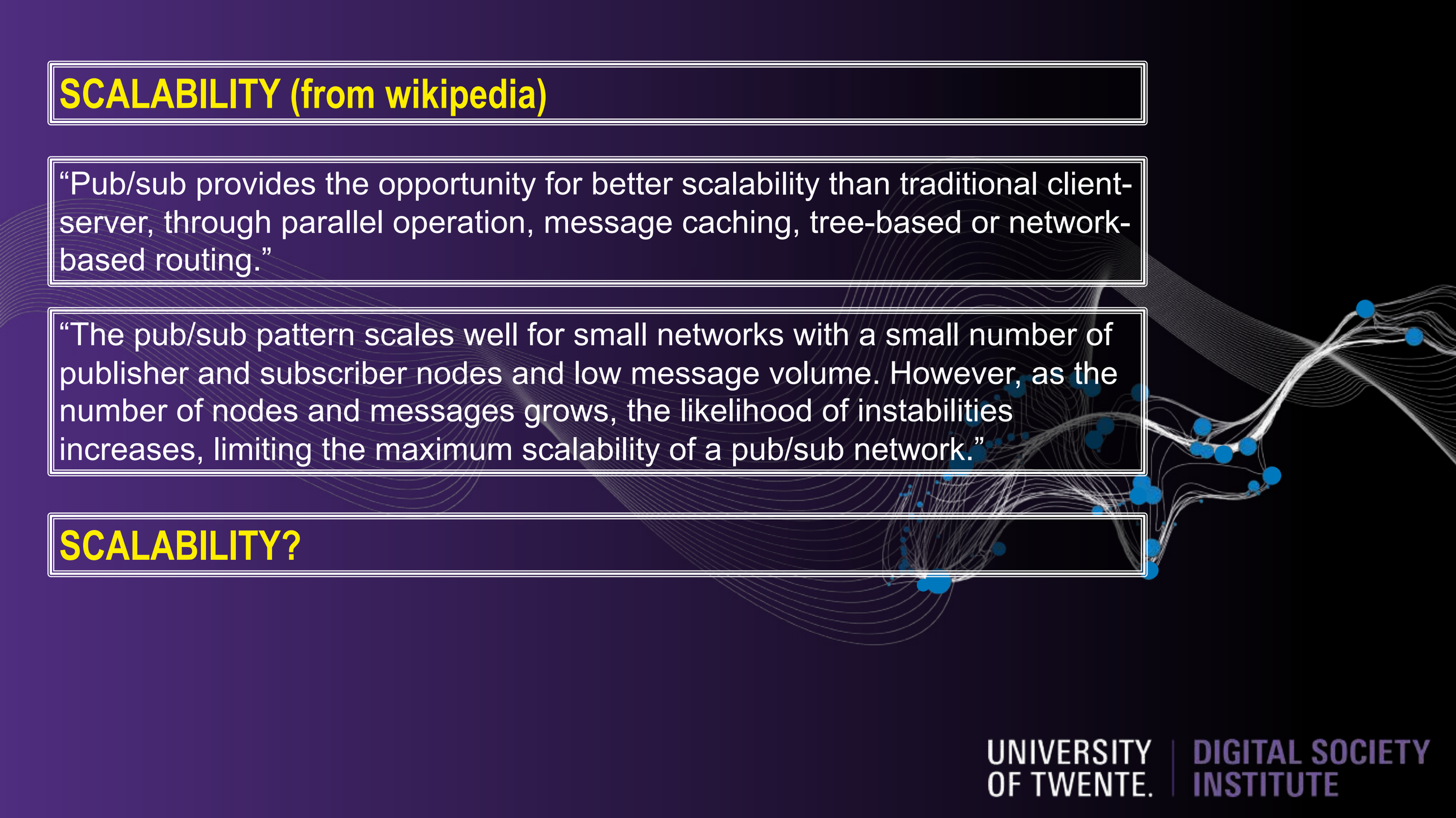
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

SCALABILITY (from wikipedia)

“Pub/sub provides the opportunity for better scalability than traditional client-server, through parallel operation, message caching, tree-based or network-based routing.”

“The pub/sub pattern scales well for small networks with a small number of publisher and subscriber nodes and low message volume. However, as the number of nodes and messages grows, the likelihood of instabilities increases, limiting the maximum scalability of a pub/sub network.”

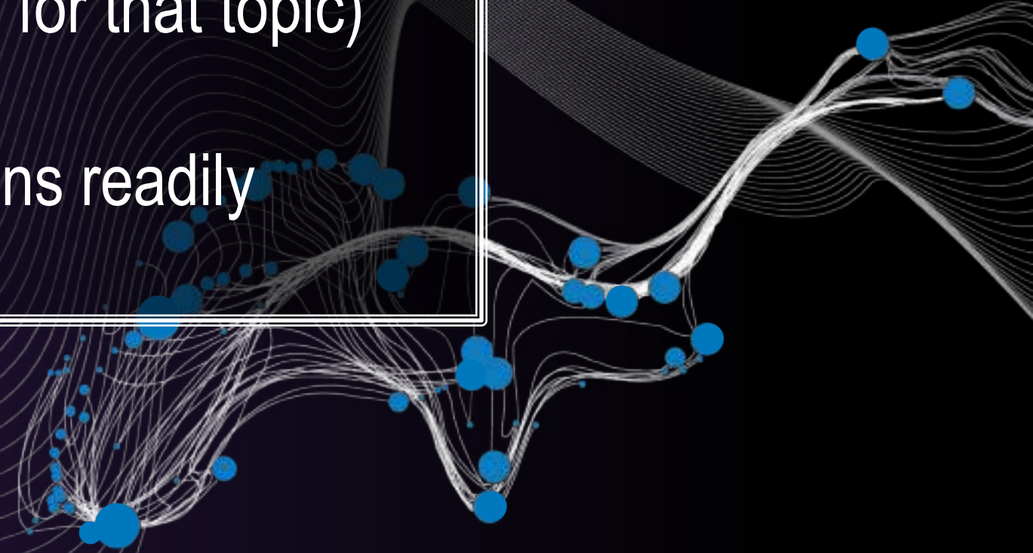
SCALABILITY?



GENERAL SOLUTIONS FOR SCALABILITY

- Restrict to topic-based pub-sub systems
- Set up different infrastructures, per topic (e.g. a single server per topic that handles all publications and subscriptions for that topic)

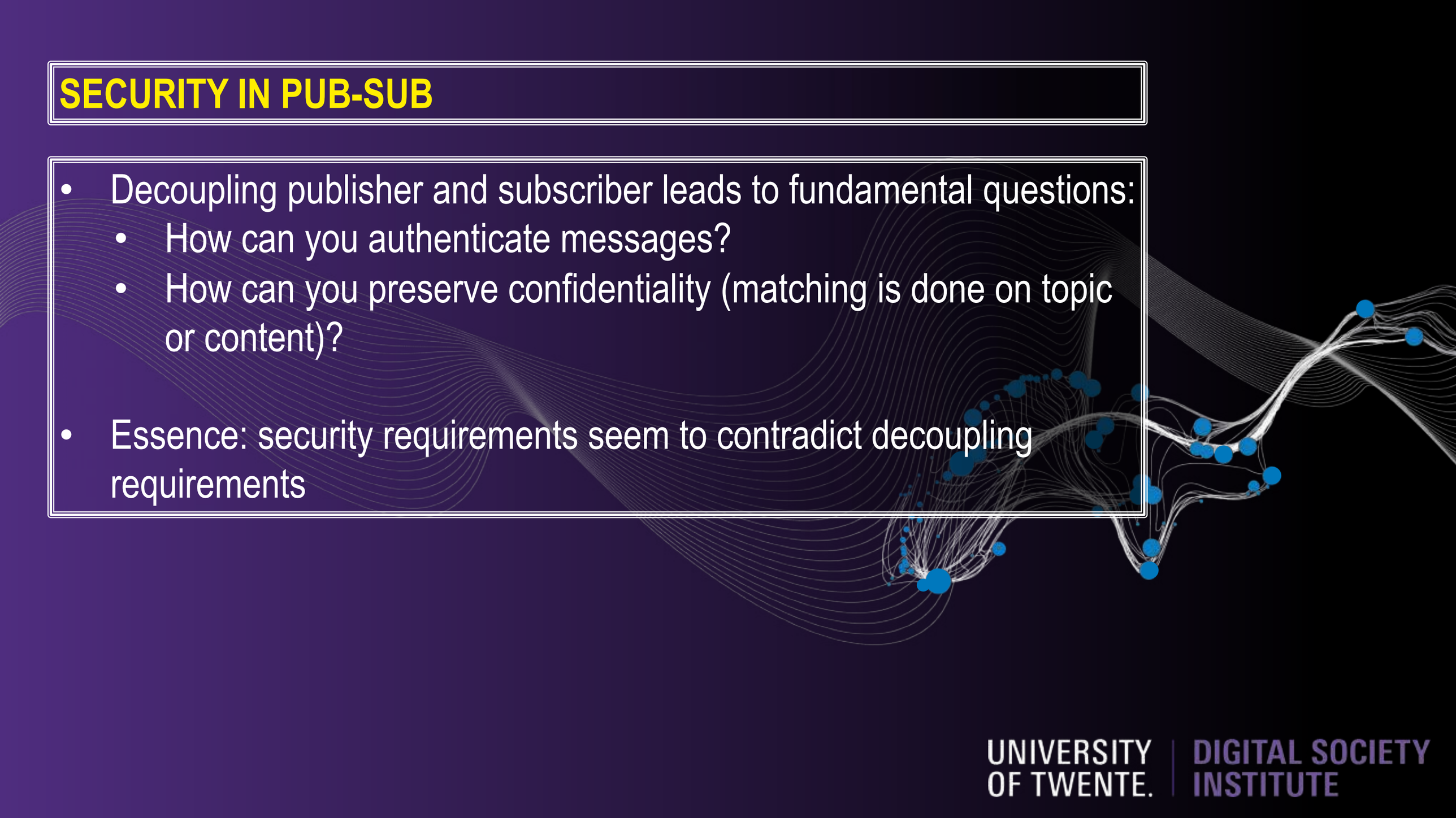
- For content-based pub-sub: no real scalable solutions readily available.



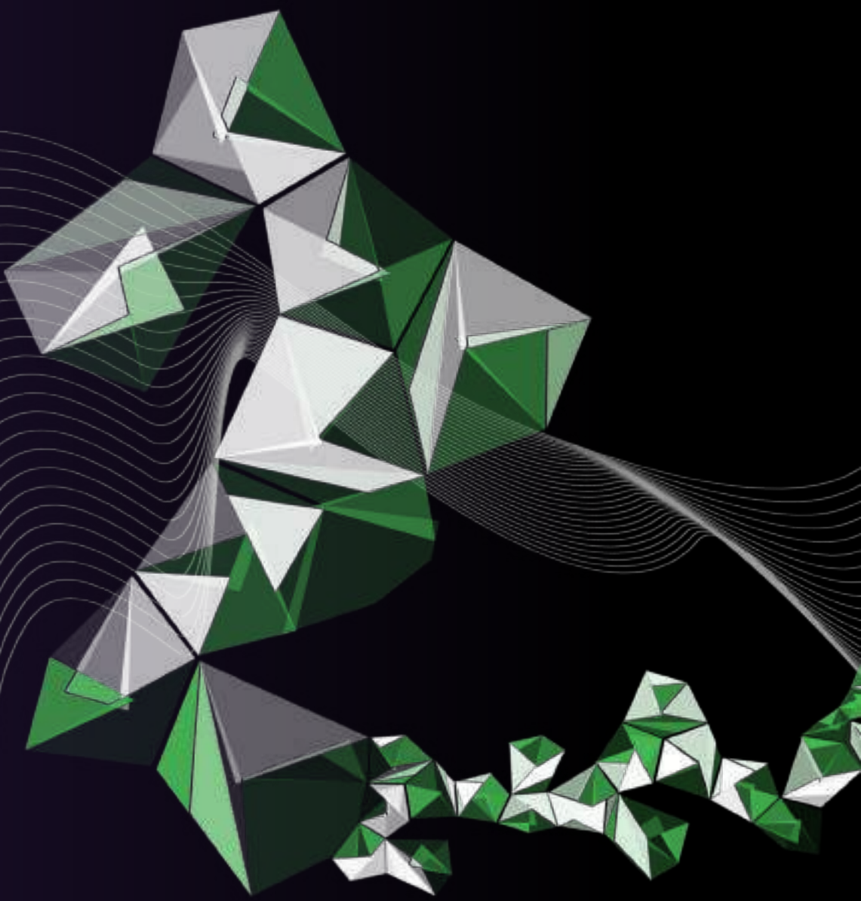
SECURITY IN PUB-SUB

- Decoupling publisher and subscriber leads to fundamental questions:
  - How can you authenticate messages?
  - How can you preserve confidentiality (matching is done on topic or content)?

- Essence: security requirements seem to contradict decoupling requirements



WORK TO DO



UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

STARTING POINTS & SUGGESTIONS

- The Many Faces of Publish-Subscribe.
  P. Eugster, P. Felber, R. Guerraoui, A.M. Kermarrec.
  ACM Computing Surveys, vol. 35(2), June 2003.

- A Survey of Security Solutions for Distributed Publish-Subscribe Systems
  A.V. Uzinov.
  Computers & Security 61 (2016).

  Both papers give a good overview of publish-subscribe systems

- Distributed Systems book
  H1, H2.1, H2.3, H4.3, H4.4, H5.1, H5.5, H6.6

- A number of other papers on scalability or security within the context of publish-subscribe will be provided upon request.