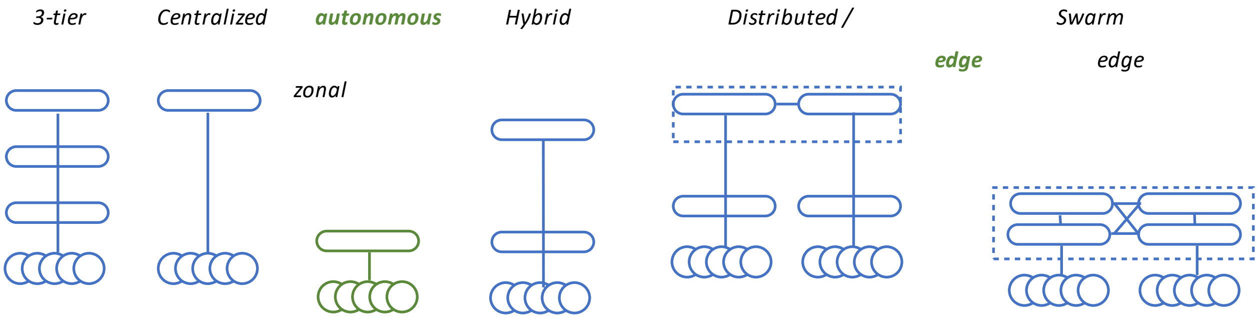
IIC Architecture - Autonomous edge pattern

Key question	Where to place information or logic?
Alternatives	Centralized, Hybrid, Swarm
Stakeholder	Service/Product designer

1. Problem

Information can be kept in the things at the edge, in a central location, distributed in all those places, replicated in diverse locations or can follow many creative schemes. In the same idea, applying logic on the information can be done at several places:



In those diagrams, rounds represent things which can be sensors, controllers. The rectangular objects represent levels/nodes of data handling and/or storage.

Distributed or Zonal pattern is a case where a number of systems collaborate to solve the problem. The presence of horizontal links intend to picture intra-level communication as opposed hierarchical communications nature of the 3-tier or centralized patterns.
The « geographic dispersion » can be between states in USA or between zones in a car (rear/left, rear/right, front/left, front/right, central1, central2).

The swarm pattern is a dynamic collection of collaborative nodes such as cars in a platoon.

Choosing one pattern can be either difficult or almost imposed by regulatory or operational constraints. Each pattern is described in a specific pattern document.

2. Description

This autonomous edge pattern presents the case where most of the value generation activity is placed at the edge. It does not exclude limited storage or compute in the things.

2.1. Solution

The autonomous edge pattern places information and/or its handling in multiple nodes located near where it is produced. Things are connected with a single edge node which apply logic, stores data and performa analytics. This edge node may be connected to a central node for data backup purposes. Multiple edge nodes may be connected to collaborate (for instance a light pole may inform what it infers will be the next light pole to turn on before the next pole detect the walking pedestrian).

Requirements to do so range from costs efficiencies to technical requirements such as latency to access the information.

The pattern is composable with « centralized » pattern into « hybrid edge » pattern: information stored at the edge may be a cached relevant subset of the full information base that may be centralized.

2.2. Model

Things produce data and sends it to the autonomous edge node it is associated. Data is stored and handled at this node. Multiple nodes are working in full independence from each other and from any central entity.

Consolidation data may be sent to a central location but are *not* part of the logic of the service. The edge nodes may be managed from a central location but can still be considered autonomous edge as the decision making is autonomous. This is the main difference with 3-tier and distributed patterns where another level of logic is involved in the handling of the received data.

2.3. Stakeholder

This pattern is intended for the service or product designer that need to weigh efficiency, costs and possible regulatory constraints when choosing an overall system pattern such as the autonomous edge

3. Pattern Guidance

3.1. Advantages

Operational constraints typically pushes the autonomous edge rather than advantages.

For instance, latency requirements are typically a reason to use autonomous edge: if latency is below 10ms, information or compute location in the cloud is currently virtually impossible.

Another factor can be regulatory constraints: a remote health care monitoring solution may face huge complexities if centralized. Keeping patient information local at his home may avoid all issues.

Scale problems may also impose autonomous edge: if an engine generates terabytes of data per day, it may be impossible or economically not viable to transport the data in a central location.

For multi-geography solutions, it may be easier to deploy localized versions of the application rather than having to handle multiple regional and language preferences on a central platform.

3.2. Disadvantages

Purchase costs and maintenance costs are proportional to service usage. A way to mitigate this is to integrate those costs in service subscription if the business model allows this.

Over-the-air updates are mandatory and increase complexity of management.

Physical exposure to attackers can become a hard and costly problem to solve.

Integration in edge computing environment may be problematic. For instance, elders may not have internet at all at the moment they want to benefit of the service. Partnerships with telecom operators may become necessary.

3.3. Other considerations

It may be a good practice to start a service with a central architecture then apply the autonomous edge as the service usage grows. If this is designed from the beginning, then the most effective way to do it is to design the entire hybrid architecture from scratch but colocate all components in a centralized location. As the service usage grows, components can be gradually pushed out to the edge.

Autonomous edge does not impose Internet connectivity. For instance an autonomous remote health services can be using telephone lines to send alerts and leverage the patient smartphone to be « bridged » to the remote health box (that is addressing security and configuration issues).

4. Application notes

See more on applying the pattern in:

- Remote health monitoring / autonomous edge
- Analytics / autonomous edge