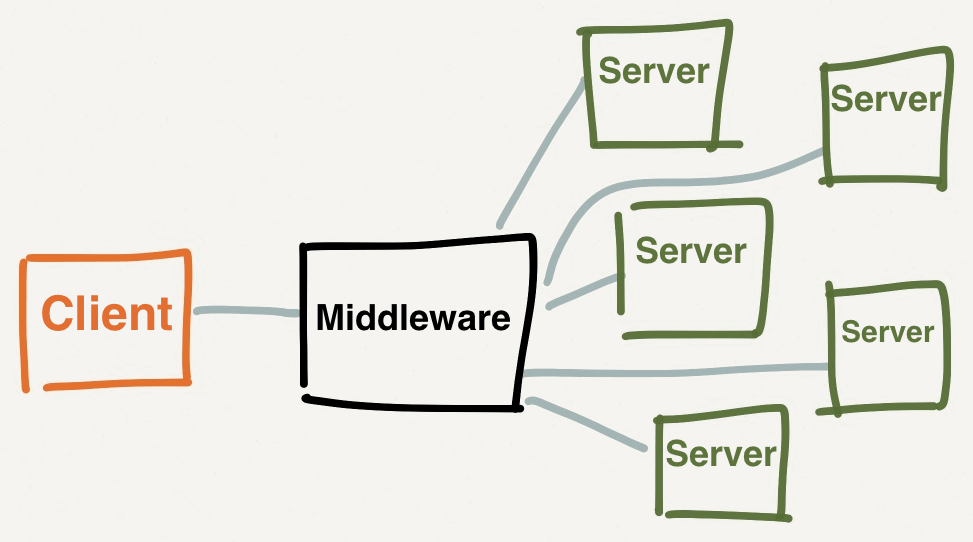
Overview of MCollective Components and Configuration

The MCollective Components

- Kiến trúc của Mcollecive dựa trên 3 thành phần chính: server, clients và middleware.

- Server và client sử dụng các thành phần khác nhau



Note:

+ Thuật ngữ Mcollective về server có 1 số khác biệt so với Puppet

+ Khái niềm về “web server” or “puppet master server” trên puppet, thay đổi bằng khái niệm “SSH server” trên mcollective. Đây là nhưng máy chủ đang thực hiện các hoạt động khác nhưng vẫn lăng nghe các request từ Mcollective

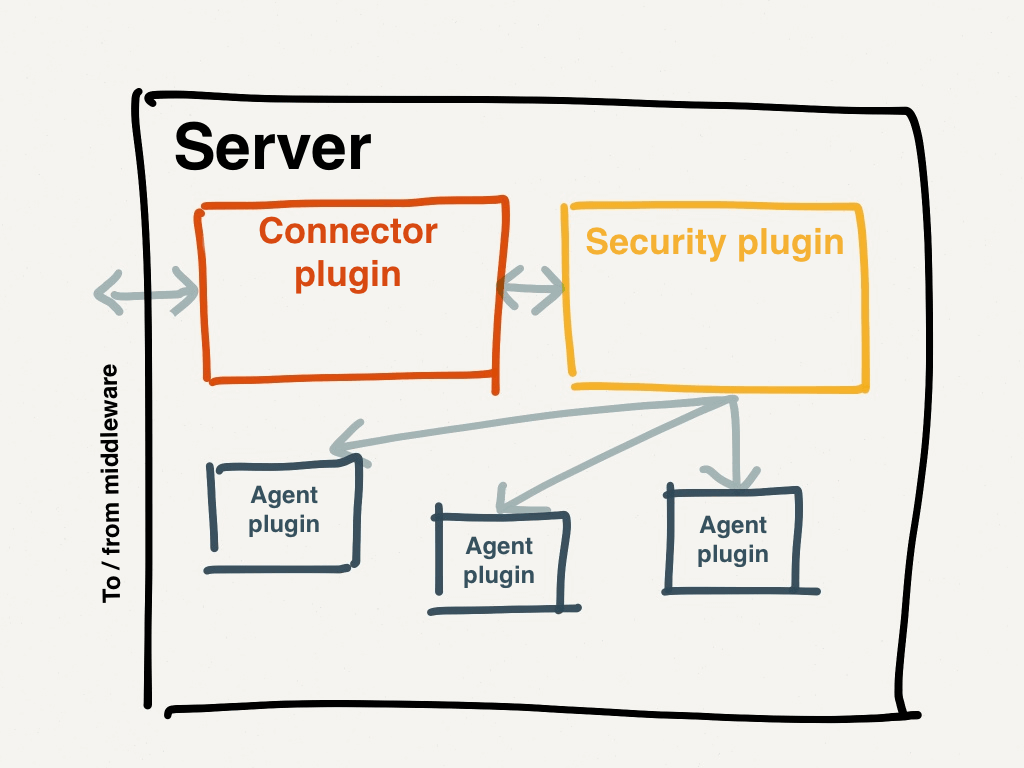
+ Từ góc nhìn người dùng, server chấp nhận các request và đáp trả. Từ góc nhìn middleware, các máy chủ cũng là 1 loại client, nó được kết nối đến middlerware và đăng ký loại message nó quan tâm

+ Thuật ngữ về agent cũng khác biệt. Trong puppet, agent là tiến trình ngầm, tìm là applies các config, nó tương đương với tiến trình mcollecived. Trong Mcollective, agent chỉ là 1 nhóm các hành động được phân phối bởi plugin

|  |  |  |
| --- | --- | --- |
| Thing | Puppet | Mcollective |
| Service that makes changes to the system | puppet agent | mcollectived server daemon |
| Plugins that enable new actions/functionality | types and providers | agent plugins |

Servers

- Mcollective server (thường gọi là “node”) là 1 máy tính có thể được điều khiển thông qua Mcollective. Server chạy Mcollective daemon (mcollectived) và có 1 lượng lơn các agent plugins được installed



- “mcollectived” service sử dụng 1 số loại plugin:

+ connector plugin để kết nối tới các middleware, thu thập các request và gửi các replies

+ security plugin cho việc kiểm tra, lọc các request, encode các replies

+ 1 vài agent plugins thực thi các action từ requests

- connector và security plugins được ủy quyền, clients cũng có thể có các plugin giống server

- Server có thể sử dụng:

+ registration plugin gửi heartbeat và các metadata tới 1 số loại inventory database

+ Data plugins cho phép lọc các request phức tạp

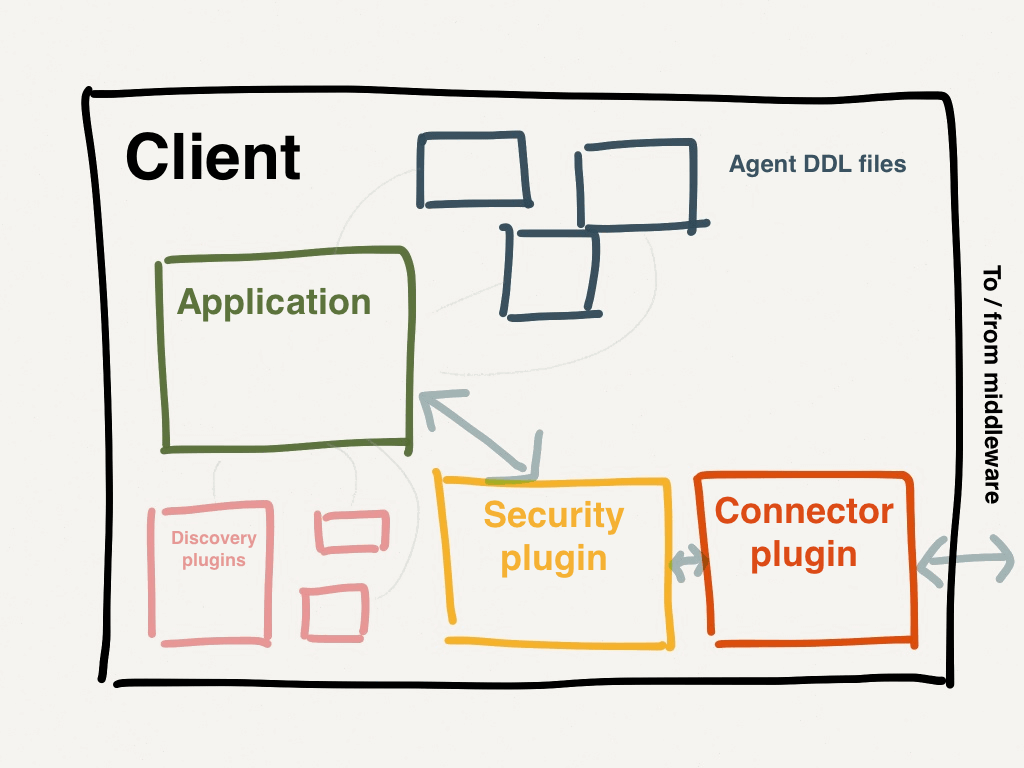
+ authorization plugin ủy quyền các xác nhận request trên mỗi hành động

+ audit plugin sử dụng để log tất cả request tại centrel or local

Clients

- Mcollective client có thể gửi request tới 1 số các server, sử dụng security plugin cho việc encode và sign request và sử dụng connectir plugin để publish nó.

- Các plugin phải match security plugin và connector used bởi servers. Client cũng nhận được các replies từ server và format các response cho mỗi user hoặc 1 số hệ thống khác.



- Các client thường gặp là mco command-line client. Nó có thể sử dụng để tương tác hoặc trong các script. Có thể write 1 số client trong Ruby như backend GUI apps hoặc glue in a reactive infrastructure.

- Giống server, client sử dụng 1 số loại plugins:

+ connector plugin để kết nối middlerware và gửi các request (subscibe to replies)

+ security plugin để validate và filter các requests (encode replies)

+ 1 vài agent plugin Data Definition Language (DDL) file cho việc xác thực các requests

- Clients có thể sử dụng

+ Extra discovery plugins cho việc get lists các server node sẽ đáp trả các request. Mặc định discovery methods sử dụng empty Mcollective message để tìm các node, querying central database

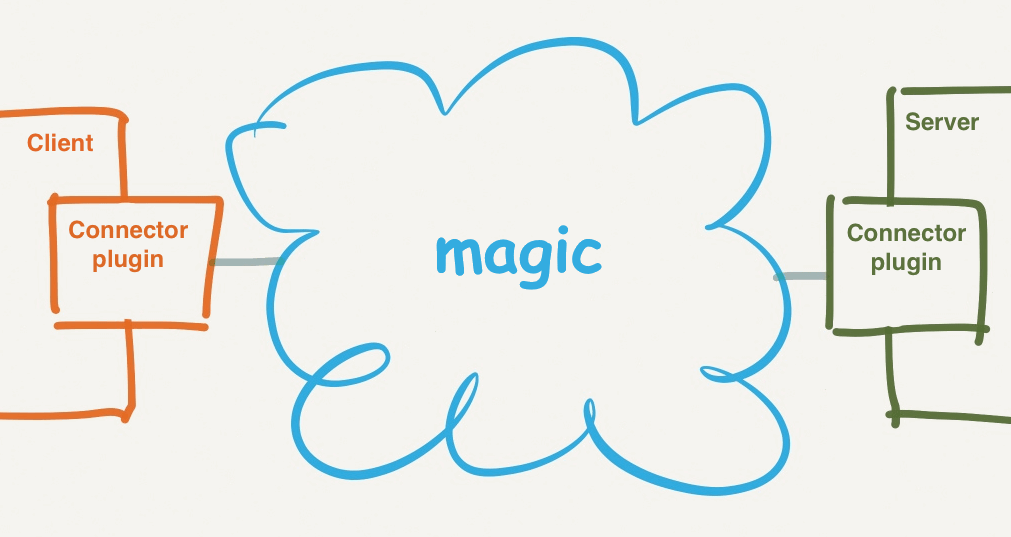
+ validator và aggergate (tập hợp) plugin để kiểm tra và format data. Nó có thể tham chiếu đến bởi các DDL files

+ Application plugins để add các custom subcommand tới mco command.

Middleware

- Mcollective client và server không kết nối trực tiếp đến nhau, chúng kết nối tới 1 số middleware system để route messages, publish message tới middleware và subscribe messages nhận lại

- middleware system ở bên ngoài Mcollective, tất các tương tác trực tiếp với nó được thực thi bởi connector plugin, nó cần 1 số thống tin về middleware topology và config. Nó sẽ trừ tượng tất cả kết nối thành 1 abstract cloud of magic:

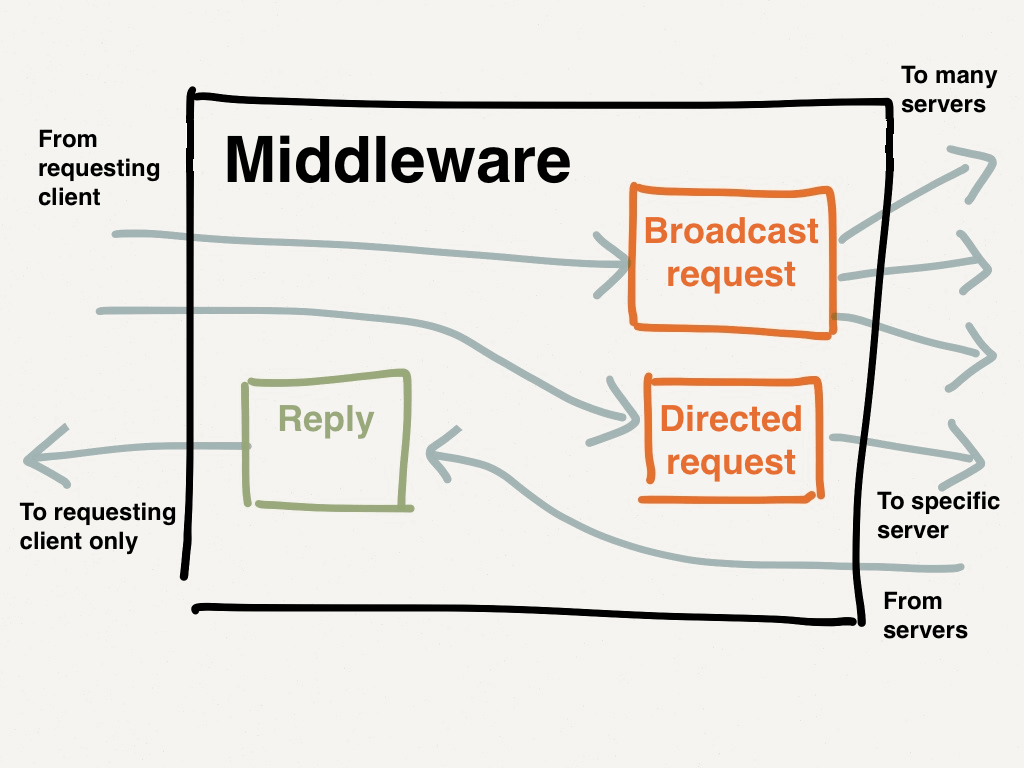


- Connector và middleware xử lý 3 loại message

+ broadcast requset (client to server)

+ Directed request (client to server)

+ Replies (server to client)



Global Configuration and Deployment Decisions

- 1 số config Mcollective là global, và cần phải match tất cả thành phần. Triển khai sẽ trở nên dễ dàng nếu hiểu về các cấu hình này

- The main kinds of shared configuration are:

+ Middleware type and connector

+ Security plugin

+ Agent plugins

+ Subcollectives

+ Authorization policies

Middleware Type and Connector

- tất cả server và client cần sử dụng các connector plugin giống nhau và nó phải match middleware type

- Bên cạnh nó, cách cấu hình middleware sẽ ảnh hưởng tới 1 số settings connecter plugin. Nó bao gồm:

+ Hostname(s) and port(s)

+ Username and password

+ Whether to use SSL

+ SSL credentials (if using CA verification)

Security Plugin

- Tất cả user và client cần sử dụng cùng security plugin. Nếu nó cần các chứng chỉ cho việc chứng thức, server sẽ phải cấu hình accept clients credentials. Mỗi security plugin sẽ có chưng yếu cầu riêng về chứng chỉ và cách phân phối

+ The SSL security plugin (recommended)

+ The PSK security plugin (only recommended for demo environments)

+ The AES security plugin (only recommended in special cases)

Agent Plugins

- Mcollective servers cần có agent plugins. Không phải tất cả server đều có chung tập agent, nhưng tất cả clients cần đẩy đủ agent DDL file để thực hiện các action.

Subcollectives

- Nếu chia site thành subcollective cho security hoặc traffic, bạn cần cấu hình đây đủ list collecive và xác định rõ các business rule để gán cho các node

- server và client cần biết tập các subscribe và gửi các request

- The middleware needs to know about them as well, since subcollectives are implemented in the connector plugin.

Authorization Policies

- Nếu bạn sử dụng authorization plugin like ActionPolicy, bạn sẽ phải cầu hình policies thích hợp trên các server.

- Rule cho mỗi policies cần match unique identities trên clients, mà sẽ được gửi các request, so the policies will be tied to your central knowledge of who and what the clients are.

- With the ActionPolicy plugin, all policy files can safely be distributed to all servers, since each one defines which servers its rules apply to based on facts and other metadata.