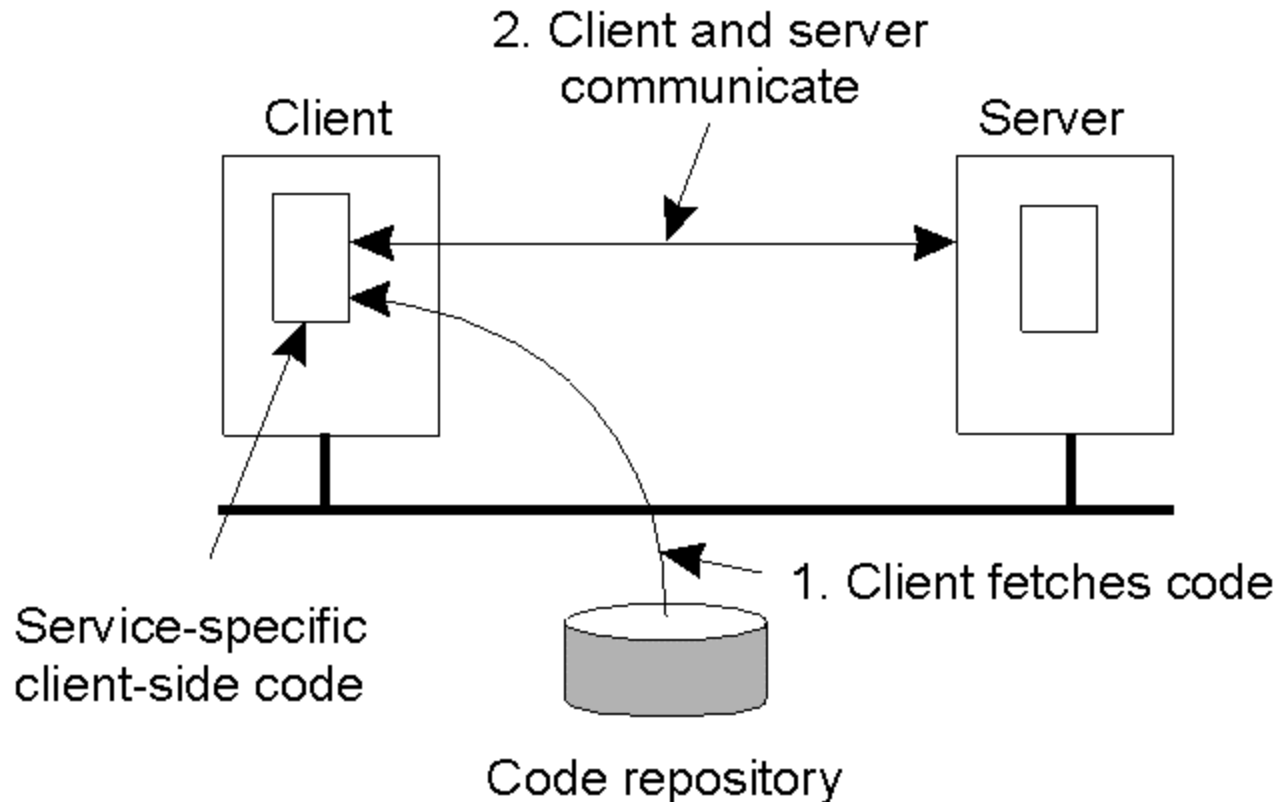# Processes

Chapter 3

Code Migration

Code migration in distributed systems:

process migration

statically or dinamically

for improving performance or for application goals.

Reasons for Migrating Code

The principle of dynamically configuring a client to communicate to a server. The client first fetches the necessary software, and then invokes the server.

Code Migration

Weak mobility: moving *code segment* and initialization data. (es. Java applet)

Strong mobility: moving *code segment, execution segment* and initialization data.

Models for Code Migration
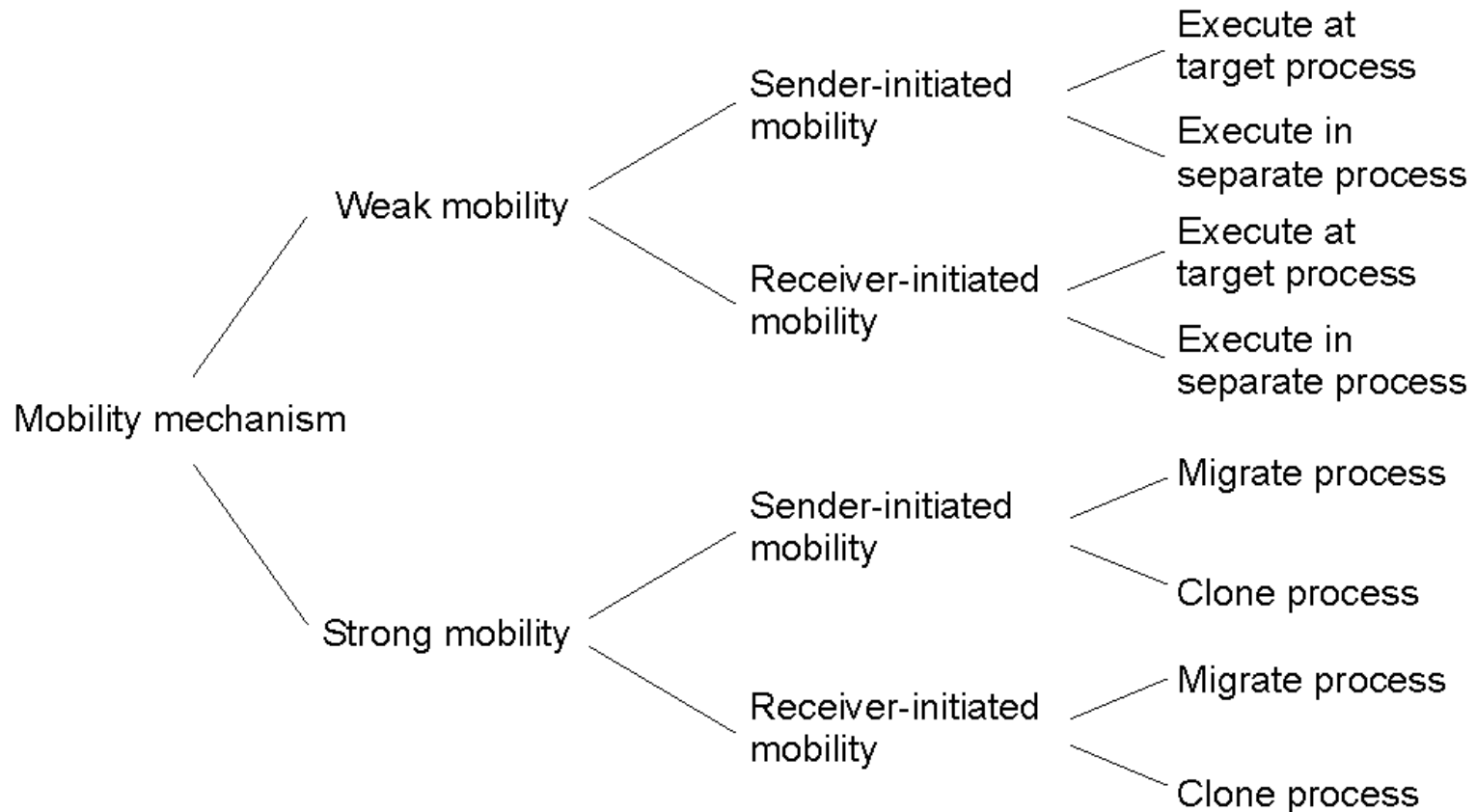
Alternatives for code migration.

Migration and Local Resources

Resource-to machine binding

	Unattached	Fastened	Fixed
By identifier	MV (or GR)	GR (or MV)	GR
By value	CP (or MV, GR)	GR (or CP)	GR
By type	RB (or GR, CP)	RB (or GR, CP)	RB (or GR)

Process-to-resource binding (left label for the three rows: By identifier, By value, By type)

Actions to be taken with respect to the references to local resources when migrating code to another machine.

Migration in Heterogeneous Systems

- Is more complex.

- Requires code portability.

- A virtual machine approach is used.

- Weak mobility is easier

- In strong mobility it is necessary to handle the execution segment.

- A migration stack is used.

Migration in Heterogeneous Systems
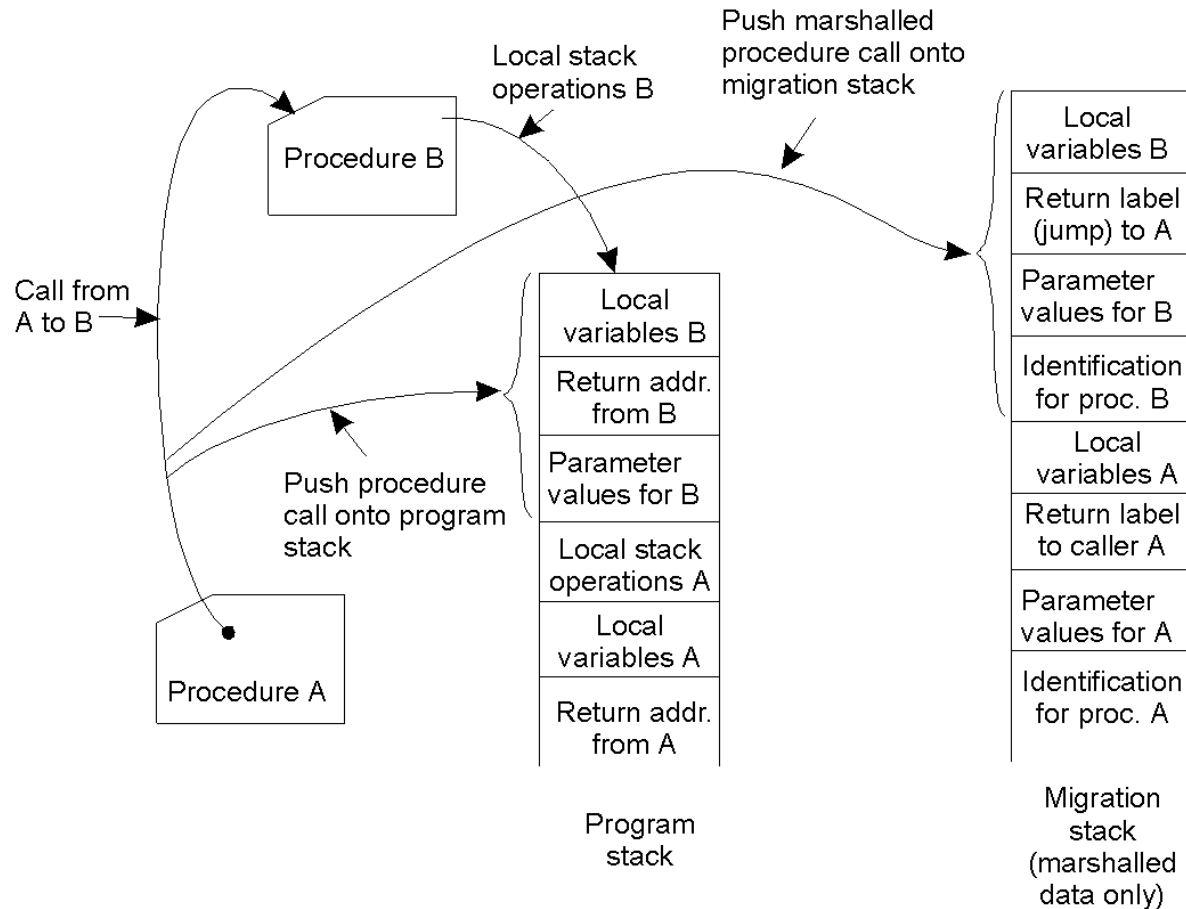
The principle of maintaining a migration stack to support migration of an execution segment in a heterogeneous environment

Software Agents in Distributed Systems

- **Software agent**: an *autonomous* process capable of reacting to, and initiating changes in *collaboration* with users and other agents.

- A software agent can take initiative.

- Mobile agents, interface agents, information agents.

Software Agents in Distributed Systems

Property	Common to all agents?	Description
Autonomous	Yes	Can act on its own
Reactive	Yes	Responds timely to changes in its environment
Proactive	Yes	Initiates actions that affects its environment
Communicative	Yes	Can exchange information with users and other agents
Continuous	No	Has a relatively long lifespan
Mobile	No	Can migrate from one site to another
Adaptive	No	Capable of learning

Some important properties by which different types of agents can be distinguished.

Agent Technology
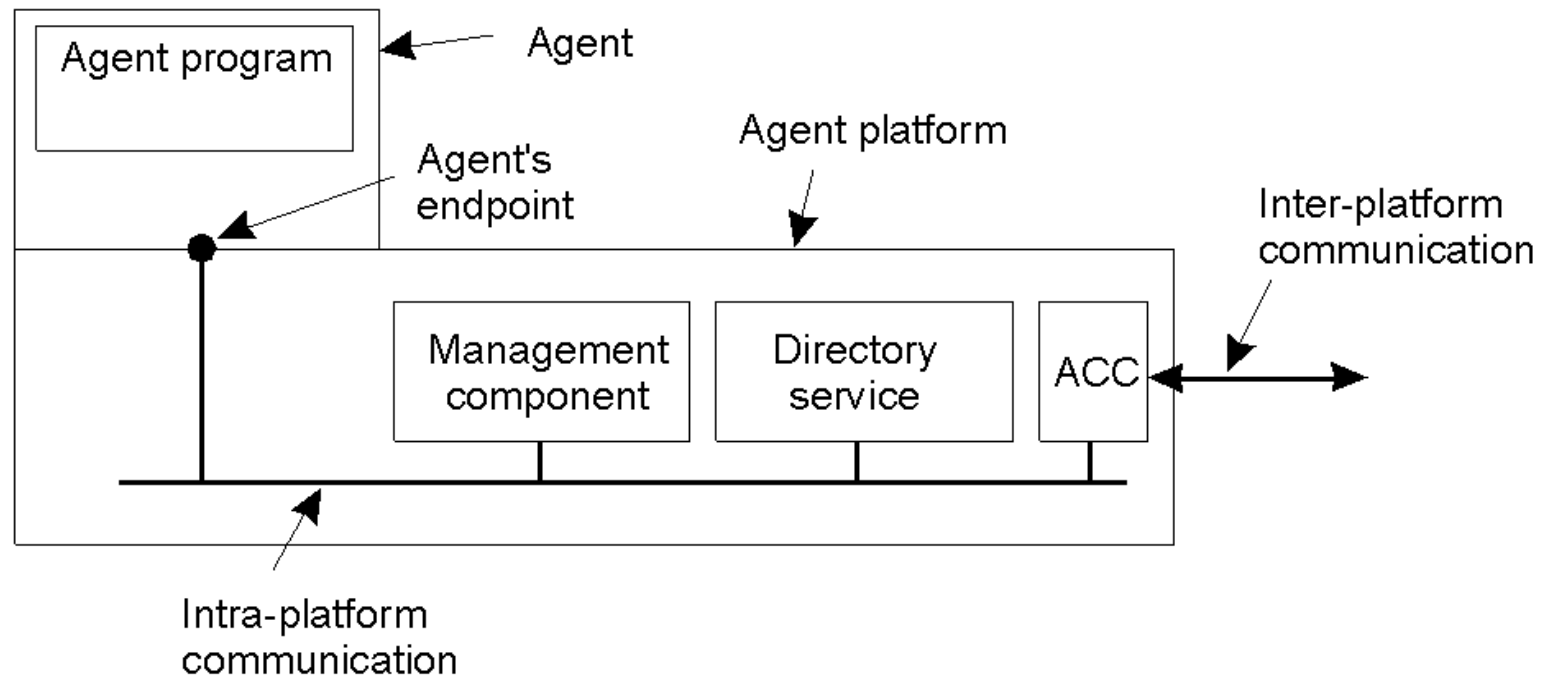
The general model of an agent platform (adapted from [fipa98-mgt]).

Agent Communication Languages (1)

Message purpose	Description	Message Content
INFORM	Inform that a given proposition is true	Proposition
QUERY-IF	Query whether a given proposition is true	Proposition
QUERY-REF	Query for a give object	Expression
CFP	Ask for a proposal	Proposal specifics
PROPOSE	Provide a proposal	Proposal
ACCEPT-PROPOSAL	Tell that a given proposal is accepted	Proposal ID
REJECT-PROPOSAL	Tell that a given proposal is rejected	Proposal ID
REQUEST	Request that an action be performed	Action specification
SUBSCRIBE	Subscribe to an information source	Reference to source

Examples of different message types in the FIPA ACL [fipa98-acl], giving the purpose of a message, along with the description of the actual message content.

Agent Communication Languages (2)

Field	Value
Purpose	INFORM
Sender	max@http://fanclub-beatrix.royalty-spotters.nl:7239
Receiver	elke@iiop://royalty-watcher.uk:5623
Language	Prolog
Ontology	genealogy
Content	female(beatrix),parent(beatrix,juliana,bernhard)

A simple example of a FIPA ACL message sent between two agents using Prolog to express genealogy information.