

# **Distributed Systems**

Third edition

Version 3.02 (2018)

(Contains minor corrections in comparison to 3.01)

Maarten van Steen  
Andrew S. Tanenbaum

**Get a free copy of this book at:**

**<https://www.distributed-systems.net/index.php/books/ds3/>**

---

# CONTENTS

---

Preface	<b>Week 1</b>	xi
<b>1 Introduction</b>		<b>1</b>
1.1 What is a distributed system? . . . . .		2
Characteristic 1: Collection of autonomous computing elements		2
Characteristic 2: Single coherent system . . . . .		4
Middleware and distributed systems . . . . .		5
1.2 Design goals . . . . .		7
Supporting resource sharing . . . . .		7
Making distribution transparent . . . . .		8
Being open . . . . .		12
Being scalable . . . . .		15
Pitfalls . . . . .		24
1.3 Types of distributed systems . . . . .		24
High performance distributed computing . . . . .		25
Distributed information systems . . . . .		34
Pervasive systems . . . . .		40
1.4 Summary . . . . .		52
<b>2 Architectures</b>		<b>55</b>
2.1 Architectural styles . . . . .		56
Layered architectures . . . . .		57
Object-based and service-oriented architectures . . . . .		62
Resource-based architectures . . . . .		64
Publish-subscribe architectures . . . . .		66
2.2 Middleware organization . . . . .		71
Wrappers . . . . .		72
Interceptors . . . . .		73
Modifiable middleware . . . . .		75
2.3 System architecture . . . . .		76

**Week 4**

Weeks  
2/3

	Centralized organizations . . . . .	76
	Decentralized organizations: peer-to-peer systems . . . . .	80
	Hybrid Architectures . . . . .	90
2.4	Example architectures . . . . .	94
	The Network File System . . . . .	94
	The Web . . . . .	98
2.5	Summary . . . . .	101
3	<b>Processes</b> . . . . .	<b>103</b>
3.1	Threads . . . . .	104
	Introduction to threads . . . . .	104
	Threads in distributed systems . . . . .	111
3.2	Virtualization . . . . .	116
	Principle of virtualization . . . . .	116
	Application of virtual machines to distributed systems . . . . .	122
3.3	Clients . . . . .	124
	Networked user interfaces . . . . .	124
	Client-side software for distribution transparency . . . . .	127
3.4	Servers . . . . .	128
	General design issues . . . . .	129
	Object servers . . . . .	133
	Example: The Apache Web server . . . . .	139
	Server clusters . . . . .	141
3.5	Code migration . . . . .	152
	Reasons for migrating code . . . . .	152
	Migration in heterogeneous systems . . . . .	158
3.6	Summary . . . . .	161
4	<b>Communication</b> . . . . .	<b>163</b>
4.1	Foundations . . . . .	164
	Layered Protocols . . . . .	164
	Types of Communication . . . . .	172
4.2	Remote procedure call . . . . .	173
	Basic RPC operation . . . . .	174
	Parameter passing . . . . .	178
	RPC-based application support . . . . .	182
	Variations on RPC . . . . .	185
	Example: DCE RPC . . . . .	188
4.3	Message-oriented communication . . . . .	193
	Simple transient messaging with sockets . . . . .	193
	Advanced transient messaging . . . . .	198
	Message-oriented persistent communication . . . . .	206
	Example: IBM's WebSphere message-queuing system . . . . .	212
	Example: Advanced Message Queuing Protocol (AMQP) . . . . .	218

4.4	Multicast communication . . . . .	221
	Application-level tree-based multicasting . . . . .	222
	Flooding-based multicasting . . . . .	226
	Gossip-based data dissemination . . . . .	229
4.5	Summary . . . . .	234
<b>5</b>	<b>Naming</b>	<b>237</b>
5.1	Names, identifiers, and addresses . . . . .	238
5.2	Flat naming . . . . .	241
	Simple solutions . . . . .	241
	Home-based approaches . . . . .	245
	Distributed hash tables . . . . .	246
	Hierarchical approaches . . . . .	251
5.3	Structured naming . . . . .	256
	Name spaces . . . . .	256
	Name resolution . . . . .	259
	The implementation of a name space . . . . .	264
	Example: The Domain Name System . . . . .	271
	Example: The Network File System . . . . .	278
5.4	Attribute-based naming . . . . .	283
	Directory services . . . . .	283
	Hierarchical implementations: LDAP . . . . .	285
	Decentralized implementations . . . . .	288
5.5	Summary . . . . .	294
<b>6</b>	<b>Coordination</b>	<b>297</b>
6.1	Clock synchronization . . . . .	298
	Physical clocks . . . . .	299
	Clock synchronization algorithms . . . . .	302
6.2	Logical clocks . . . . .	310
	Lamport's logical clocks . . . . .	310
	Vector clocks . . . . .	316
6.3	Mutual exclusion . . . . .	321
	Overview . . . . .	322
	A centralized algorithm . . . . .	322
	A distributed algorithm . . . . .	323
	A token-ring algorithm . . . . .	325
	A decentralized algorithm . . . . .	326
6.4	Election algorithms . . . . .	329
	The bully algorithm . . . . .	330
	A ring algorithm . . . . .	332
	Elections in wireless environments . . . . .	333
	Elections in large-scale systems . . . . .	335
6.5	Location systems . . . . .	336

GPS: Global Positioning System . . . . .	337
When GPS is not an option . . . . .	339
Logical positioning of nodes . . . . .	339
6.6 Distributed event matching . . . . .	343
Centralized implementations . . . . .	343
6.7 Gossip-based coordination . . . . .	349
Aggregation . . . . .	349
A peer-sampling service . . . . .	350
Gossip-based overlay construction . . . . .	352
6.8 Summary . . . . .	353
<b>7 Consistency and replication</b>	<b>355</b>
7.1 Introduction . . . . .	356
Reasons for replication . . . . .	356
Replication as scaling technique . . . . .	357
7.2 Data-centric consistency models . . . . .	358
Continuous consistency . . . . .	359
Consistent ordering of operations . . . . .	364
Eventual consistency . . . . .	373
7.3 Client-centric consistency models . . . . .	375
Monotonic reads . . . . .	377
Monotonic writes . . . . .	379
Read your writes . . . . .	380
Writes follow reads . . . . .	382
7.4 Replica management . . . . .	383
Finding the best server location . . . . .	383
Content replication and placement . . . . .	385
Content distribution . . . . .	388
Managing replicated objects . . . . .	393
7.5 Consistency protocols . . . . .	396
Continuous consistency . . . . .	396
Primary-based protocols . . . . .	398
Replicated-write protocols . . . . .	401
Cache-coherence protocols . . . . .	403
Implementing client-centric consistency . . . . .	407
7.6 Example: Caching and replication in the Web . . . . .	409
7.7 Summary . . . . .	420
<b>8 Fault tolerance</b>	<b>423</b>
8.1 Introduction to fault tolerance . . . . .	424
Basic concepts . . . . .	424
Failure models . . . . .	427
Failure masking by redundancy . . . . .	431
8.2 Process resilience . . . . .	432

Resilience by process groups . . . . .	433
Failure masking and replication . . . . .	435
Consensus in faulty systems with crash failures . . . . .	436
Example: Paxos . . . . .	438
Consensus in faulty systems with arbitrary failures . . . . .	449
Some limitations on realizing fault tolerance . . . . .	459
Failure detection . . . . .	462
8.3 Reliable client-server communication . . . . .	464
Point-to-point communication . . . . .	464
RPC semantics in the presence of failures . . . . .	464
8.4 Reliable group communication . . . . .	470
Atomic multicast . . . . .	477
8.5 Distributed commit . . . . .	483
8.6 Recovery . . . . .	491
Introduction . . . . .	491
Checkpointing . . . . .	493
Message logging . . . . .	496
Recovery-oriented computing . . . . .	498
8.7 Summary . . . . .	499
<b>9 Security</b>	<b>501</b>
9.1 Introduction to security . . . . .	502
Security threats, policies, and mechanisms . . . . .	502
Design issues . . . . .	504
Cryptography . . . . .	509
9.2 Secure channels . . . . .	512
Authentication . . . . .	513
Message integrity and confidentiality . . . . .	520
Secure group communication . . . . .	523
Example: Kerberos . . . . .	526
9.3 Access control . . . . .	529
General issues in access control . . . . .	529
Firewalls . . . . .	533
Secure mobile code . . . . .	535
Denial of service . . . . .	539
9.4 Secure naming . . . . .	540
9.5 Security management . . . . .	541
Key management . . . . .	542
Secure group management . . . . .	545
Authorization management . . . . .	547
9.6 Summary . . . . .	552

<b>Bibliography</b>	<b>555</b>
---------------------	------------