1. Record Nr. UNINA9910449876703321

Autore	Dyson Paul
Titolo	Architecting enterprise solutions [[electronic resource]] : patterns for high-capability Internet-based systems / / Paul Dyson, Andy Longshaw
Pubbl/distr/stampa	Hoboken, N.J., : John Wiley & Sons, 2004
ISBN	1-280-27034-9 9786610270347 0-470-85587-8
Descrizione fisica	1 online resource (384 p.)
Collana	Wiley Software Patterns Series
Altri autori (Persone)	LongshawAndy
Disciplina	005.2/76
Soggetti	Internet programming Computer architecture Computer systems Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Contents; Acknowledgements; An All-Too-Common Story; In the beginning ...; In the middle ...; In the end ...; Chapter 1 Introduction; What this Book is About; What this Book is Not About; Why Write this Book?; Who Should Read this Book?; Architects; Developers; Project Managers; Students and Trainees; The Structure of the Book; Part 1; Part 2; Part 3; Reading the Book; PART 1 Architecture, Patterns and Internet Technology; Chapter 2 System Architecture; Architecture, Design and 'Goodness of Fit'; How does this Relate to System Architecture? What are the Non-functional Characteristics we Care About?Balancing the Non-functional Characteristics; Defining System Architectures; Why do we Care About System Architecture?; Summary; Chapter 3 Internet Technology Systems; Types of Internet Technology System; Why do we Build Systems Using Internet Technology?; Building Blocks of Internet Technology Systems; Why is it Difficult?; Summary; Chapter 4 Architectural Patterns for Internet Technology Systems; Patterns, Languages and Internet Technology; Why do we use Patterns?; Patterns vs Principles; Organization of the Patterns

Presentation of the Patterns Navigating the Language; Summary; Chapter 5 The GlobalTech System; The Business Case; The System Overview; Summary; PART 2 The Patterns; Chapter 6 Fundamental Patterns; Getting the Shape Right; APPLICATION SERVER ARCHITECTURE; PERIPHERAL SPECIALIST ELEMENTS; What makes these Patterns Fundamental?; Summary; Chapter 7 System Performance Patterns; I Feel the Need, the Need for Speed; Principles; ACTIVE-REDUNDANT ELEMENTS; LOAD-BALANCED ELEMENTS; SESSION FAILOVER; DEDICATED WEB AND APPLICATION SERVERS; COMMON PERSISTENT STORE; Why Tiers are not a Catastrophe
DATA REPLICATION CONNECTION LIMITATION; RESOURCE POOLING; LOCAL CACHE; OFFLINE REPORTING; Other Patterns; Chapter 8 System Control Patterns; Dangerous (adj.): Speed Without Control; Terminology; CONTINUAL STATUS REPORTING; OPERATIONAL MONITORING AND ALERTING; 3-CATEGORY LOGGING; SYSTEM OVERVIEW; DYNAMICALLY-ADJUSTABLE CONFIGURATION; DEMILITARIZED ZONE; INFORMATION OBSCURITY; SECURE CHANNELS; Secure Sockets Layer and SSL Acceleration; KNOWN PARTNERS; Other Patterns; Chapter 9 System Evolution Patterns; Plus Ca Change; Principles; DYNAMICALLY-DISCOVERABLE ELEMENTS; EXPANDABLE HARDWARE
VIRTUAL PLATFORM SWAPPABLE STAGING ENVIRONMENT; SEPARATE SYSTEM-MANAGED DATA; Other Patterns; PART 3 Application of the Patterns; Chapter 10 GlobalTech Revisited; Reviewing the Architecture; Architecting for System Performance; Architecting for System Control; Architecting for System Evolution; Summary; Chapter 11 Applying the Patterns; Not Quite the Simplest System that Could Work; Which Patterns to Apply; A Process for Applying the Patterns?; Examples of Applying the Patterns; Summary; Chapter 12 Moving on from Here; Technology; Tool Support; Development Process
An Evolutionary Approach to Architecture

Sommario/riassunto

A practical, nuts-and-bolts guide to architectural solutions that describes step-by-step how to design robustness and flexibility into an Internet-based systemBased on real-world problems and systems, and illustrated with a running case studyEnables software architects and project managers to ensure that nonfunctional requirements are met so that the system won't fall over, that it can be maintained and upgraded without being switched off, and that it can deal with security, scalability, and performance demandsPlatform and vendor independence will empower architects to