学校代码：10246
学　　号：12212010038

硕 士 学 位 论 文

（专业学位）

移动电子履历系统的设计与实现

Design and Implementation of

Mobile Electronic Pedigree System

院　　系：　软件学院

专　　业：　软件工程

姓　　名：　彭蔚蔚

指 导 教 师：　韩伟力 副教授

完 成 日 期：　2014 年 10 月 10 日

指导小组成员名单

韩伟力 副教授

目录

摘要

电子履历技术可以保障物联网中物流系统数据的可信度，因而被工业界和学术界所关注。尽管国内外已有一些电子履历相关的系统，并被应用到药品等高附加值商品的追溯中，但这些系统没有应对当前高速发展的移动互联网的技术特性。即，这些系统的客户端能力很弱，不能满足更多产品如食品的追溯需求。本论文旨在研究如何设计并实现一个兼顾移动便携、易用性、安全性并且能够实际应用的移动电子履历系统，以满足不断扩大的安全追溯需求。

本论文设计了一个基于 Android 平台的移动电子履历系统，实现了产品在整个供应链流程中的信息化管理，从而便于快速追溯，及时定位问题，提高消费者对产品的信任度。该系统通过基于 Android 平台的移动设备，采取二维码识别或其它感知方式，收集、存储并处理产品在各个环节上的信息，遵循设计制定的规范，生成相应的电子履历文件，并且利用数字签名技术进行履历签署，以保障履历的可信性。同时，该系统向移动端提供了验证和展示产品电子履历的接口，方便消费者快速地查询履历并全面地获取产品相关信息。本论文进一步针对移动互联网应用在电子履历系统中的挑战，对系统进行性能优化，并且采取一些安全措施对数据进行保护，从而使得移动电子履历较传统电子履历系统而言，更加方便快捷易用。

关键字：物联网，电子履历，移动互联网，Android

Abstract

Electronic pedigree systems recently obtain much attention from industry and academia, as they can be deployed to ensure the trustworthiness of data in logistic systems in the Internet of Things. Many of such systems have been developed to track high additional value commodities, such as drugs, but few of them leverages the developing mobile internet technologies. That is, the client of these systems are weak and cannot meet the requirements of tracking varies kinds of products, such as food. This paper is motivated to design and implement a practical mobile electronic pedigree system that works on mobile platform, and provides the features of both ease of use and security to meet the increasing demand of secure tracking.

This paper designs an electronic pedigree system based on the Android platform that makes it easy to tracing conveniently and quickly, and locating problem timely by applying electronic information management in the product supply chains, and enhance the consumers' confidences in products. The proposed system collects, stores and processes all the information of the product using an Android device with sensing methods, such as QR code recognition, and generates the corresponding electronic pedigree following our proposed specification. The generated electronic pedigree is signed using certification to guarantee the credibility. We provides APIs for the verification and presentation of electronic pedigree in a mobile client device, to make it easier for customers to query the information of products. This paper further discusses the challenges of mobile electronic pedigree applications, and improves the efficiency and security of our system to make it much easier to use compared to the conventional electronic pedigree systems.

Keywords: Internet of Things, Electronic Pedigree, Mobile Internet, Android

第一章 绪论

1.1 研究背景

近几年来，食品、药品、农产品等产品的安全问题备受瞩目。这些安全事故的频频发生，引起了消费者的恐慌，降低了社会对于相关产业的信赖度。2004 年阜阳劣质奶粉事件[1]以及 2008 年奶制品三聚氰胺污染事件[2]，导致婴幼儿致病致死，造成了我国民众对于整个奶制品行业的不信任。2013 年在台湾发生了一系列重大食品安全问题事件[3]，由于涉及到的食品与生活息息相关，故极大地降低了台湾民众对食品安全的信任度，从而引发了社会的高度关注和讨论。2012 年央视曝光的毒胶囊事件[4]也影响广泛，社会反响强烈，使得民众对整个药品行业深深担忧。这些安全事件的接连发生，促使人们寻找一种解决方法，能够保障产品的安全性，便于快速追溯，及时定位问题，提高消费者对产品的信任度。

而随着物联网的高速发展，越来越多的物联网技术应用到了各个领域中，实现电子化、智能化的信息管理[34]，因而面向产品的电子履历系统应运而生。利用物联网技术，电子履历系统能够方便、快速地定位产品中的安全问题，对产品在整个供应链中的流程进行标签追溯，提高产品的可信任度和安全性。基于物联网的电子履历系统[5]利用计算机安全手段，针对活动在物联网中的产品的生命周期各个环节的信息进行记录、保存和签名，从而实现可追溯和可信任。目前，国内外对于电子履历系统都具备了一定的研究，并在小范围内进行了实施。但当前的研究和实施都还只是一些初步应用，存在一些问题，在规范通用性、方便易用、安全性、可信任性、流程完备性等方面仍有一些不足，具有一定提升的空间。

与此同时，随着智能手机、平板等设备和移动应用的普及，移动互联网蓬勃发展，各式各样的应用深入人们的生活，改善了人们的生活质量，将时代推向移动互联网时代。根据 IDC（国际数据公司，International Data Corporation）的统计报告[6]，仅仅 2013 年第四季度一季度，Android 智能手机的销量就达到 2 亿余部，占据市场总额的 78.1%。同时，Android 应用也在飞速发展，据 AppBrain 应用市场统计[7]，截至 2014 年 6 月 30 号，Android 官方市场上的应用程序数量已达到 1,316,773。

而移动智能终端的感知、处理能力的快速发展以及应用程序的普及，为移动互联网与物联网的结合提供了机会，将物与物以及人与物的交互关系更紧密地整合在了一起，形成一种庞大而联系紧密的网络。而对于移动物联网方面，目前的研究还不够成熟，实际应用也不够广泛，缺乏行之有效的解决方案[39]。

对于物联网的应用代表之一电子履历系统来说，与移动端的结合势必会带来移动性、感知能力、便携性、网络传输等应用优势和终端优势，与此同时，移动

应用的安全性以及移动端的性能限制也是亟待解决的问题[41]。而本论文的研究目的和重点就是结合移动端和电子履历系统，在充分体现两者各自的优势的同时，解决结合带来的一些问题，从而实现一个基于 Android 平台的兼顾易用性及安全性的移动电子履历系统。

1.2 国内外研究现状

自从美国麻省理工学院（MIT）的 Kevin Ashton 教授于 1999 年首次提出物联网的概念后，物联网逐步吸引了社会的关注，特别是在近几年来发展迅猛，一跃成为当今研究最为炙手可热的方向之一[8]。物联网是指利用感知识别技术与传感器等装置对物理世界的物体的信息进行收集，通过网络进行数据传输，经过数据分析与挖掘、计算和处理，实现物与物、人与物的连接和通信[9]。物联网的高速发展为电子履历的研究提供了坚实的理论基础。基于物联网所设计的电子履历系统，可以快速方便地对产品进行溯源、追踪和管理，检测产品的安全问题，使供应链的管理透明化、信息化[10]。

目前为止，国外已经提出了一些电子履历相关的技术和系统，并且取得了一定成果[32]。

EPCglobal 是一个由国际物品编码协会（EAN）和美国统一代码委员会（ UCC ）联合成立的机构，它针对药品监管提出了较为完整规范的电子履历技术[11]。该技术根据药品供应链的特性设计了履历规范，并且结合了 RFID 技术[33]，通过可信性保障手段对药品的流动轨迹进行记录和签名，从而保障各个阶段信息的完整性和可信性。该履历规范定义了药品在整个流通过程中的数据的记录，记录包含了药品信息和每一次流动分发环节相关信息，整个流程包括：药品从制造商处生产出来并进行出售，经历了各个分销商和包装商的买入、包装和售出，直到最后上架到药店以供消费者购买。经过对这些信息的验证核实后，生成一个完整的履历。通过该履历规范，可以保证药品的流程记录在案，当发生安全问题时可以快速追踪，便于追责。

此外，国外已有一些实际应用的系统。日本提出了一项关于食品安全管理的新制度，要求对食品进行电子化管理，对生产、加工、销售、流通等的各个阶段建立电子履历，以便有据可查，出问题时快速反应，快速定位原因。

德国利用条形码技术，对国内的鸡蛋进行监管，在每一个鸡蛋上都贴有红色条形码标记，其所包含的信息解码后，包括三部分：母鸡的饲养方式、鸡蛋的出产国家以及母鸡所属养鸡场和鸡舍或鸡笼的编号。从而使得消费者可以放心挑选并购买，并且一旦有鸡蛋质量出现问题，政府可以根据条形码上的信息快速地对相关养鸡场进行追责，并迅速追踪并收回已经流通到市场上的问题鸡蛋，有效防止事故进一步扩大。

在国内，对电子履历技术的研究才刚刚起步。目前电子履历技术在台湾的发展相对深入一些，这得益于台湾行政院农业委员会提出了《产销履历制度》并实施了大力推动。农委会选定了畜禽等多项农产品，推动台湾良好农业规范（TGAP: Taiwan Good Agricultural Practice）的实施，从而对农产品的安全性进行验证和保障，并且建立了履历追溯体系，以便快速有效地进行风险管控，并降低风险发生时造成的危害。在政策的推动下，目前台湾已经有多家农产品相关企业响应政府号召，建立了各自公司的农产品履历系统[12][13][14]。并且，台湾行政院农委会也建立了台湾农产品安全追溯资讯网（TAFT）[15]，进一步推动并宣传农产品履历追溯技术。

针对之前奶制品的问题频发的现象，中国味全公司搭建了针对其公司奶制品的产品履历查询系统[16]，成为国内奶制品首例。消费者可以输入奶粉包装上的履历代码进行履历查询和浏览，追踪奶粉从奶源产地到加工生产，再到上架到市场中进行销售的整个流程，从而保证奶粉的质量安全。

由国内外的研究现状可见，尽管目前已经有一些电子履历技术和系统，但这些电子履历系统都还是初步的应用，缺乏统一、规范的标准，加上大部分系统都是针对特定公司的特定产品，不具有通用性，并且对于安全性考虑得还不够全面，甚至有些系统只能通过在网站上输入文本（比如一串数字编号）的形式进行查询，不够方便易用。

另外，目前电子履历还没有和移动端进行结合的可行方案，在这方面的研究还处于比较空白的阶段。尽管移动端自身具有移动性、便捷性、易用性、感知能力、网络传输等终端优势和应用优势，但如何将其与电子履历系统相结合并发挥其优势，仍需要进一步研究。并且，考虑到移动端设备在存储能力、计算能力等性能方面的局限性，以及移动应用程序在安全性方面发展的不够成熟，存在一些安全威胁和漏洞；而电子履历系统在履历生成、履历解析等方面需要消耗一定的计算能力，同时履历的保存也会占用一定存储空间，更重要的是履历系统对安全性有较高的要求，故如果要将移动技术应用到电子履历系统，还有很多问题亟待解决。

由此可见，由于电子履历系统和移动端各自的特性，移动电子履历系统的设计和实现还存在很大的挑战。本论文的研究需要在结合移动端和电子履历系统的过程中，展现两者的优势，同时尽量解决结合带来的一些问题。

1.3 研究内容及意义

本论文的研究目的是设计一个基于 Android 平台的移动电子履历系统，实现产品在整个供应链流程中的信息化管理，从而便于快速追溯，及时定位问题，提高消费者对产品的信任度。该系统通过基于 Android 平台的移动设备，采取二维

码识别或其它感知方式，针对产品在整个供应链上各个环节的信息进行收集、保存和处理，遵循设计制定的规范，根据事件信息生成相应的电子履历，并且利用数字签名技术进行履历签署，以保障履历的可信性。同时，该系统向移动端提供了验证和展示产品电子履历的接口，方便消费者快速地查询履历并全面地获取产品相关信息。

本课题的研究意义是将物联网与移动手持设备相结合，使得电子履历系统能够应对当前高速发展的移动互联网的技术特性，设计并实现一个基于 Android 移动端的农产品电子履历系统，兼顾移动便携、易用性、安全性，并且能够应用到实际中。

本论文的主要贡献如下：

- 基于 Android 平台设计并实现了移动电子履历系统，遵循设计制定的履历标准，完善业务流程，实现了产品在整个供应链流程中的信息化管理，并通过数字签名技术保障信息的完整性和可信性。
- 结合了物联网和移动互联网，充分分析了电子履历系统与移动端和移动应用的特性来进行设计和实现，提供多维感知方式，使得移动电子履历系统较传统电子履历系统而言，更加方便快捷易用。
- 针对移动互联网应用在电子履历系统中的挑战，分析具体的功能特性和数据特性，对系统进行性能优化，并且采取一些安全措施对数据进行保护。

1.4 论文的组织结构

本篇论文一共分为五章：

第一章：绪论。介绍了食品、药品等安全事故的频频发生促使人们寻求解决方法，伴随着飞速发展的物联网技术，电子履历系统应运而生，再加上移动互联网的蓬勃发展和移动端、移动应用的普及，引出移动电子履历系统的研发意义。同时对国内外研究状态进行分析和比较，确定研究内容、目标和意义。

第二章：相关技术综述。介绍和移动电子履历系统相关的技术。主要介绍物联网、电子履历系统、Android 平台等相关概念和技术，为移动电子履历系统的研发提供基础。

第三章：移动电子履历系统的分析。介绍移动电子履历系统的主要需求分析以及业务流程分析，对参与者和功能需求进行了分析，分析了产品供应链的业务流程，以及移动端系统的实际应用场景，并面向移动进行了相应的分析。

第四章：移动电子履历系统的设计。介绍系统的总体架构和各个主要模块的设计，包括：履历生成和查询、信封生成和检查等，还介绍了数据模型，包括：履历的种类和属性，信封的结构等，并且设计了各个主要功能模块的具体实现流

程，还针对移动端的特性提出了性能优化及安全保护方面的改进措施。

第五章：移动电子履历系统的实现。介绍了移动电子履历系统的开发环境配置，以及具体代码的实现，展示了应用的界面实现，介绍了系统的部署。

第六章：总结与展望。总结全文，总结本论文的主要工作，列举了所面临的挑战及下一步的工作，展望移动电子履历系统的应用前景。

第二章 相关技术综述

2.1 物联网

物联网概念的正式提出还要追溯到 1999 年在美国召开的移动计算和网络国际会议上，由 MIT Auto-ID 中心的 Ashton 教授在研究 RFID（射频识别，Radio Frequency Identification）时提出[17]。之后物联网受到了越来越多的关注，迅速地发展起来。2009 年 9 月，欧盟委员会信息和社会媒体司 RFID 部门负责人 Lorent Ferderix 博士在“物联网与企业环境中欧研讨会”上，代表欧盟提出了对物联网的定义，即：物联网是一个动态的全球网络基础设施，基于标准规范和通信协议，对物理世界和虚拟世界中的物体赋予了物理属性、虚拟特性、身份标识和智能化接口，从而实现与信息网络的无缝整合。在未来，物联网将与企业互联网、服务互联网以及媒体互联网一起搭建新的互联网时代[18]。

物联网作为新时代信息技术的重要组成部分之一[19]，通过 RFID 技术以及激光扫描器、红外传感器、全球 GPS 定位系统等传感网技术以及条形码等技术，遵循统一标准的规范协议，可以很方便地将物理世界中的任何物体接入虚拟网络世界，进行信息交换和通信，比如物体的移动、环境的数据等信息，从而实现对物体的感知识别、追踪定位、实时监控和管理[25]。物联网的概念是以互联网为核心和基础的，利用先进的传感技术以及数据传输和数据处理技术，形成一个局部应用网络，使得位于该网络中的物理物体能被感知、识别，通过对其本身性质的解析和对不同物体之间关系的理解，构建全球性的网络。

物联网主要通过三个关键技术实现了物与物相连、人与物互连：传感器技术、网络通信技术和信息处理技术，分别通过感应处理终端、传输通道、控制处理平台进行实施[37]。通过传感器技术让本来没有生命的东西变得智能化，能够被感知并识别到，相当于自动开口说话，再通过网络通信技术将信息传输到指定的地方或人，相当于人可以和远处的物体进行对话，同时物体与物体之间也能进行交流，进一步地，还可以通过控制处理命令等反过来进行控制和指挥，从而实现物理世界真正的连通。

物联网的应用前景广泛，可以应用到智能交通、智能家居[40]、医疗药品流通、社会公共安全、工业制造生产、环境预测监控（特别是恶劣地区的环境监测和灾害预测）等多个领域，从而提高人们的生活水平。

一个标准的物联网系统由四个技术架构层面所构成：感知识别层、网络构建层、管理服务层和应用层[36]，如图 2.1 所示。

感知识别层位于物联网四层架构的最底层，提供了所有上层架构的基础，主要负责通过安置在物理物体上的传感器或者阅读器，利用短距离通讯技术和协同

信息处理，感知并采集物体周围的环境信息等，形成一定规模的传感网络。现实世界中的各种静态、动态信息，例如物体的温度、轨迹、速度等基本信息，都可以借助感知识别层中的传感器（比如压力传感器、声音传感器、温度传感器、光强传感器等）、RFID 技术、二维码或条形码技术等来进行数据采集。在采集到所需的信息数据之后，感知识别层通过简单的短距离数据传输对这些数据整合处理，并提取出有效信息提供给上层架构[26]。

图 2.1 物联网技术架构[10]

网络构建层主要负责向上层管理服务层传输感知到的信息，同时向下层感知识别层传输命令，从而作为感知识别层和管理服务层之间的连接纽带。其数据的传输需要用到当前的多种网络形式，包括各种本地网络、互联网、有线和无线通信网、移动通信网络等。网络构建层需要保证数据传输的安全可靠，还面临很多技术挑战，例如怎样保证传送的信息快速完整地传输、怎样对每一个物联网中的物体进行识别等。

管理服务层负责对获取到的大量感知数据进行储存、计算、分析，实现数据的有效整合和利用[38]。主要解决数据如何存储、如何检索、如何使用、如何不被滥用等问题，相应地，用到的技术包括数据库与海量存储技术、搜索引擎、数据挖掘与机器学习、数据安全与隐私保护等。

应用层位于整个四层架构的最上层，具有丰富的物联网应用，提供整个物联网与用户的接口，使得用户可以进行方便高效的操作，实现物联网的智能化应用。这些应用以物体或者物理世界为中心，涉及到智能家居、智能物流、智能交通、智能电力、物品追踪、环境感知和检测、工业监控等各个领域。目前，物联网应

用还在快速发展中，数量正在迅速增长，具有行业化、规模化、多样化的特性[24]。

2.2 电子履历

电子履历借鉴了现实世界中履历的基本概念，通过可信性保障手段对活动在物联网中的物品的轨迹进行记录和签名，从而保障物流系统的信息的完整性和可信性。其表现形式是一种记录产品履历信息并附有数字签名的文件。

一般来说，电子履历系统是一个利用履历形式记录产品供应链上各环节信息的系统[30]。即对于每件产品，都会附加一份电子履历，该电子履历记录了该产品从生产到流通最后到销售整个供应链过程中的信息，并且每个环节的信息都具有生成该信息的厂家的数字签名。消费者只需要通过产品的唯一序列号就可以在线查看验证产品的电子履历，从而获取产品的所有履历信息。

随着物联网的蓬勃发展，越来越多的物联网技术（主要是 RFID）[35]应用到了实际中，使得电子履历系统有了初步的发展，特别是在药品行业应用了较为成熟的电子履历系统。RFID 技术[25]引起美国 FDA（食品药品监督管理局，Food and Drug Administration）了兴趣和关注。美国 FDA 在 2004 年时建议将 RFID 应用于药品追踪，形成药品电子履历来保障药品的安全性，作为防止伪造药品的解决方案之一。此外，加利福尼亚州药物管理部首次推出了一项应用电子履历的法令，要求该州内所有正在运营的处方药产业在规定年限前应用电子履历系统这一技术，该电子履历既可以采用条形码又可以采用射频识别技术来完成。

随着电子履历相关技术研究的进行，EPCglobal 在 2007 年提出了一份关于药品电子履历的规范[11]。该规范将药品电子履历被定义为一种经过签名验证的数据记录，包含药品的每一次分发的信息，从而体现药品在整个供应链中流动的信息。该规范针对药品从被制造商生产、流通至分销商和包装商、最后在药店销售的整个过程，定义了初始履历、重新包装履历、附加履历、发货履历、收货履历以及不签署发货履历。基于这一规范，IBM，Oracle 以及 SupplyScape 三家公司分别开发了各自的面向药品行业的电子履历系统。

IBM InfoSphere Traceability Server (ITS)[20]是一个易于部署的用于供应链中物品跟踪追溯的信息共享平台，可以对实时可见的事件进行创建或销毁。该系统为通过传感网收集到的供应链数据提供了可扩展的、安全的、基于标准的数据存储。企业可以使用 ITS 来捕获、管理产品相关的数据，并将其共享给内部或外部管理应用，从而进行实时分析并做出相关决策。IBM 的 ITS 产品完全遵从 EPCIS（电子产品代码信息服务）标准，实现了基于策略的安全，具有稳定性、可拓展性、可部署性等特性。

Oracle Pedigree and Serialization Manager (OPSM)[21]是一个集成的大规模序列化和履历应用的系统，使得公司可以对药物产品进行大规模序列化实施，并在

整个供应链中共享序列化的产品数据。OPSM 是专门为医药行业建立的，结合序列化和履历管理为一体，集成了应用分发的合规性和商业价值。OPSM 作为一个独立的系统，无论使用它的企业本身运行何种 ERP（企业资源计划系统，Enterprise Resource Planning）软件，OPSM 都能无缝地嵌入到企业环境中，从而给这些制药品分销公司带来一系列的好处：OPSM 的实施对 ERP 现有的交互影响很小；OPSM 可以跨越多个 ERP 系统；OPSM 使用过程中产生的大量数据不会对 ERP 系统的性能造成影响。

SupplyScape E-Pedigree 是一个是建立在 EPCglobal 网络模型之上的安全网服务系统，可以为公开供应链中的药品分销公司提供强有力的业务支持，可以面向产品进行有序编码，做出决策，整合生产流程。该系统利用原产地电子标签技术、识别技术、编码技术，为客户搭建安全可信的产品供应链，从而保证药品流通过程中的安全性。

在电子履历系统中，证书是验证履历文件签名有效性的重要凭据。其作用是绑定一个公钥和一个使用者（例如，人、服务器或设备）。ITU-T X.509 (ISO|IEC 9594-8)标准定义了一种证书格式、一种扩展机制和一组授权的扩展[29]。证书配置文件定义了实现和使用证书时必需的或者可选的元素和扩展。EPCglobal 通过 EPCglobal Certificate Profile 定义了履历使用 X.509 证书。X.509 是一种被广泛使用的数字证书标准，是国际电联电信委员会（ITU-T）为了实现单点登录和授权管理基础设施而制定的 PKI（公钥基础设施，Public Key Infrastructure）标准。该标准中定义了公钥证书、属性证书、证书销毁清单和证书路径校验算法等。本论文中的电子履历使用了这种证书。

在电子履历系统中，通过对履历文件进行数字签名来保障履历文件的真实性和有效性[28]。W3C XML-Signature Syntax and Processing 定义了一种使用 XML（可扩展标记语言，eXtensible Markup Language）创建和描述数字签名的方法[27]。本论文中履历文件中的数字签名使用了这种标准。履历中的数字签名必须遵照上文提到的证书配置文件，并采取支持 RSA 算法的签名方法和支持 SHA1 算法的 Digest 认证方法。对于数字签名的确认，系统必须满足证书配置文件、核心认证以及签名者证书认证。

2.3 Android 平台

2.3.1 Android 平台的优势

目前市场上存在 Android、iOS、WinPhone、Symbian、BlackBerry 等移动平台。本论文选择基于 Android 平台设计并开发移动电子履历系统，主要是因为 Android 平台具有如下优势：

1) 数量庞大的用户群和稳居第一的市场占有率。第一部装载 Android 智能操作系统的手机发布于 2008 年 10 月份，此后在不到三年的时间内，Android 手机的市场份额迅速超越 BlackBerry、iOS 等主流智能手机操作系统，跃居为全球第一，以绝对优势成为使用人数最多的智能手机操作系统。根据 IDC 的统计，仅仅 2013 年第四季度一季度，Android 手机的销量就达到 2 亿余部，占总市场分额的 78.1%[6]。
2) 开放的系统和平台。Android 系统使用了 Google 自定义的 Linux 内核，是真正完全开源、免费的移动平台；基于该内核，Android 使用 Java 编程语言实现了系统框架。所有 Android 的源代码均是开放的，因此 Android 可以方便地被部署到各种硬件平台之上，这使得各国的手机硬件厂商和运营厂商纷纷加入到 Android 阵营中来，同时，Android 的开放性为开发者提供了便利。
3) 成熟、便捷的应用开发套件。Google 向 Android 开发者提供了 Android SDK（软件开发工具包，Software Development Kit）来方便 Android 应用的编译、调试和部署，开发者可以选择使用开源软件 Eclipse 或是 Google 提供的 Android Studio 来进行 Android 应用的开发。与 iOS 相比，Android 开发环境是跨平台的，这使得开发 Android 相比之下变得更加便捷。

并且，因为实际应用中本论文的系统需要大量地投入到生产线上使用，故需要大量装载本系统的移动设备，考虑到满足本论文系统条件的 Android 智能设备成本较低，性价比较高，很适合大规模地投放，故选择 Android 平台进行系统开发。

2.3.2 Android 平台介绍

Android 系统是基于 Linux 的开源操作系统，一般被装载于智能手机或是平板电脑上。最早由 Andy Rubin 开发，并于 2005 年由 Google 公司收购注资，并在其后由 Google 公司通过 Apache 开源许可的方式发布了其源代码。

Android 系统的架构采用了分层、结构化的设计[23]。如图 2.2 所示，共分为如下几个部分：

1) **Linux 内核**

Android 在 2.6 版本 Linux 内核的基础之上，增添了许多为移动设备定制的功能特性进行扩展，作为其内核。内核提供了 Android 操作系统的基本功能，包括文件系统、内存管理、进程管理、硬件驱动、电源管理等。此外，Android 使用的内核还提供了 Binder 机制用于进程之间的通信。

2) **标准库**

标准库作为对内核功能的拓展，提供了许多 Android 必备的功能，如：图层管理、2D 渲染引擎、SQLite 数据库、媒体库、字体库、压缩库、Webkit、SSL 和 libc 等。

3) **本地代码和 Dalvik 虚拟机**

Android 利用沙箱技术来进行应用程序间的隔离，在每个沙箱内部，Android 会启动一个 Dalvik 虚拟机的实例，在 Dalvik 内部可以执行被编译为 dex 可执行格式的 Java 代码，绝大多数的 Android 系统和应用的代码都是利用 Java 实现的；除此之外，Android 还允许应用进程在沙箱内直接执行编译后的（C/C++）二进制文件，从而允许系统和应用通过高效的本地代码，实现性能更好、功能更强大的应用或服务。Dalvik 虚拟机允许 Java 代码通过 JNI (Java Native Interface）同本地代码通信。

Dalvik 还为 Android 应用提供了运行时环境的支持，包括 Java 的标准类库中的绝大多数功能以及 Android 核心 API，Android 通过这些 API 来创建、初始化以及管理一个 Android 应用的生命周期。

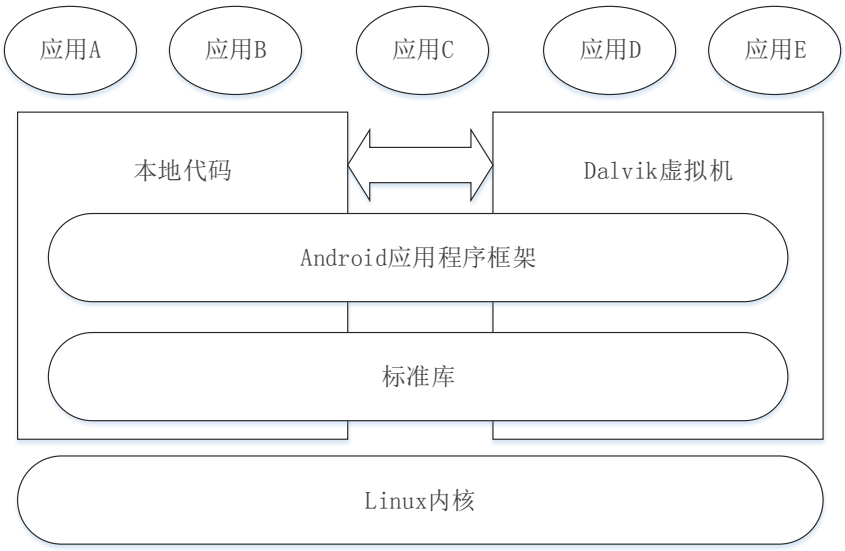

图 2.2 Android 平台架构图[23]

4) **Android 应用程序框架（Framework）**

Android 应用框架实现了 Android 系统绝大多数的逻辑功能，是构成 Android 系统的主要部分。Android 应用框架提供的功能包括：应用程序生命周期管理，包括 Android 应用的创建、挂起、恢复、销毁等，并通过继承和重写的方式将接口暴露给开发者，极大地简化了开发者的工作量；提供消息队列和进程间通信接口，在方便开发者进行多线程和多进程编程的同时，较好的规范了应用程序的结构，保证应用高效的执行；提供各种系统服务，如应用管理、通知管理、电源管理、存储管理等；封装硬件抽象层接口，使的上层应用可以通过受限的 API 与硬件交互。

Android 应用框架构成 Android 系统主体，同时搭建了 Android 应用基本架构，使得开发者可以以一种事件驱动的方式来进行 Android 应用的开发，降低了开发门槛，使得 Android 应用开发更加直观、便捷。

5) **Android 应用程序**

Android 作为一个手机操作系统，可以支持应用程序的执行，除了各种各样的第三方应用之外，移动设备的一些基本功能也是通过应用的形式来实现的，这些包含在 Android 系统内部的内置应用提供了通话与拨号、联系人、短信、Web 浏览器及多媒体浏览等众多基本功能。

2.3.3 Android 技术简介

Android 应用往往由一个或多个基本的组件构成，根据其功能不同，Android 提供了四种基本组件：Activity（活动）、Service（服务）、BroadcastReceiver（广播）和 Content Provider（内容提供者）[22]。其中 Activity 是最为常用的基本组件，一个组件可以通过创建一个 Intent（消息对象）来要求另一个组件执行特定的行为，从而实现组件间通信。下面分别对这四种组件进行简单的介绍：

1) Activity（活动）

Activity 向用户展示了一个可交互的应用界面，通过该界面用户可以实现各种操作，如拨号、收发邮件、查看地图等。通常情况下，一个应用会包含多个用户界面，也就需要多个 Activity，不同的 Activity 之间可以相互跳转，并通过 Intent 来传递数据。

一个 Activity 的生命周期在四个状态之间切换：活动状态（Resumed）、暂停状态（Paused）、停止状态（Stopped）和销毁状态（Destroyed），并向开发者提供了 onCreate()、onStart()、onResume()、onPause()、onStop()和 onDestroy()六个回调函数的入口，以便响应应用程序不同的阶段。当 Activity 启动时，首先进入活动状态，此时如果被另一个 Activity 打断，且新的 Activity 占据了整个屏幕，当前 Activity 不再可见，则当前 Activity 切换至停止状态，否则进入暂停状态，当该 Activity 再次可见并获得焦点时，切换回活动状态。当 Activity 处于停止或暂停状态时（后台执行），都有可能被操作系统强行关闭以释放内存和节省电量，处于活动状态的 Activity 也可能根据用户的操作而主动退出，此时 Activity 处于销毁状态。

Android 框架通过一个栈结构来管理同一个应用内不同的 Activity。从应用的初始 Activity 开始，应用程序每启动一个新的 Activity，之前的 Activity 都会被暂停（Pause）或停止（Stop），而新创建的 Activity 则被加入到后退栈（Back Stack）的栈顶，成为当前新的活动 Activity；当新的 Activity 完成了其生命周期或是用户点击了“后退”按钮，栈顶 Activity 会被从栈中弹出并销毁（Destroy），新的栈

顶 Activity 则会继续执行（Resume）。

2) Service（服务）

Service 的行为和 Activity 相类似，都可以作为应用程序的主要组件。与 Activity 不同，Service 不提供可视化的用户界面，而是以后台执行的方式存在，因此其生命周期较为简单，只有运行和销毁两个状态，然而 Service 的启动方式不止一种，即可以通过 startService()方法来创建一个独立运行的 Service，该 Service 只有其自身或是其创建者显示的调用 stopService()之后才会结束，也可以通过 bindService()方法启动一个临时的 Service，值得注意的是多个客户端可以绑定同一个 Service，在这种情况下，当所有的客户端都显示的调用了 unbindService()之后，临时的 Service 会被系统销毁。

Service 通常用来实现不需要用户界面、可以长时间位于后台执行的任务，例如播放音乐，处理网络、磁盘等异步 I/O。

3) BroadcastReceiver（广播）

BroadcastReceiver 用来接收特定类型的系统广播，从而使得应用程序可以获知并相应系统事件，如安装新应用、收到新短信等。从本质上来讲，Android 中的广播是一种隐式 Intent，与显示 Intent 不同，隐式 Intent 不明确指定某一个接收者，而是把该 Intent 发给所有的应用，由接收端的 Intent 过滤器决定是否处理该 Intent。通过 sendBroadcast()或 sendOrderedBroadcast()接口，系统或应用可以发送某个广播给所有注册监听某个事件的 BroadcastReceiver，从而实现一种全系统的通信。

4) Content Provider（内容提供者）

由于 Android 使用了沙箱机制，不同的应用之间无法通过文件系统共享数据，为此 Android 引入了 Content Provider，使开发者方便管理 Android 应用内的数据存储和共享，例如使普通的应用程序可以（在权限满足的条件下）访问系统联系人数据库。

Content Provider 规定了一套类似 SQL 语法的接口，提供 Insert、Delete、Update、Query 等函数供其他应用组件调用，通过这种方法，Android 实际上制定了统一的数据存储访问协议，不论是访问应用自身的本地存储还是其他应用存储的数据，都可通过 Content Provider 的接口实现。另一方面，在统一的接口下，具体的实现可以视情况而定，Content Provider 的开发者可以选择文件系统、SQLite 数据库甚至云端来进行实际的数据存储，对 Content Provider 的调用者而言，具体的存储方式是透明的。

第三章 移动电子履历系统的分析

3.1 功能需求分析

本系统是一个基于Android平台的移动电子履历系统，利用移动物联网技术，实现农产品生产、运输以及销售整个流程中信息化管理并保障信息的可信性，并具有多维感知能力和数据安全保护。它提供的主要功能是：处于生产线上的工作人员可以通过手持移动设备，根据业务系统中产生的事件信息，查看、验证、生成并上传具有数字签名的电子履历文件；企业内部使用者或者消费者可以通过开放的查询接口，对产品的履历文件进行查看。本系统的参与者包括工作人员以及消费者，具体需求分析见下文。

3.1.1 参与者分析

3.1.1.1 工作人员

工作人员需要针对系统需求，提供必要的人工操作，以保障系统的正常运营。每位工作人员都需要事先在电子履历子服务器端进行成功注册，并且在移动设备上安装移动电子履历应用程序。工作人员需要在移动端进行合法登录，身份验证通过后方可进行相关业务管理操作。

初始化移动端系统时，工作人员需要设置移动端与子服务器端的连接，连接成功方可进行相关操作。且工作人员能在之后对其设置进行修改。

处于产品生产线的工作人员，在合法登录后需要向该移动电子履历系统输入相关的生产线数据，进行一系列操作，包括：二维码识别，履历的读取、验证与生成，履历信封的读取、检查与生成等。

3.1.1.2 消费者

消费者指的是购买或使用该系统相关的产品的客户。所有消费者可以通过移动电子履历系统对产品履历进行查询。

当消费者购买产品后，在移动设备上安装并打开移动电子履历应用程序，无需进行登陆，就可以在联网的情况下，通过摄像头扫描产品上的二维码，或者通过按键输入序列号信息，查询到产品的电子履历信息。移动端的应用将通过网络从服务器端获取该产品的履历信息，并将该产品从生产到流通至今经过的所有流程，以倒叙的方式，直观地呈现在手机应用界面上。

3.1.2 用例图分析

如图 3.1 所示，描述了移动电子履历系统的各个用例。可见系统的参与者包括消费者和工作人员。其中与消费者相关的用例只有电子履历浏览，工作人员相

关的用例包括系统设置、登录、电子履历管理和信封管理。电子履历管理包括电子履历浏览和电子履历生成两个模块，其中电子履历浏览可以通过文本输入或者通过二维码扫描进行查询浏览；信封管理包括信封读取、信封检查、信封生成三个模块。二维码查询和信封读取都需要用到二维码识读。具体的用例需求分析见下一节。

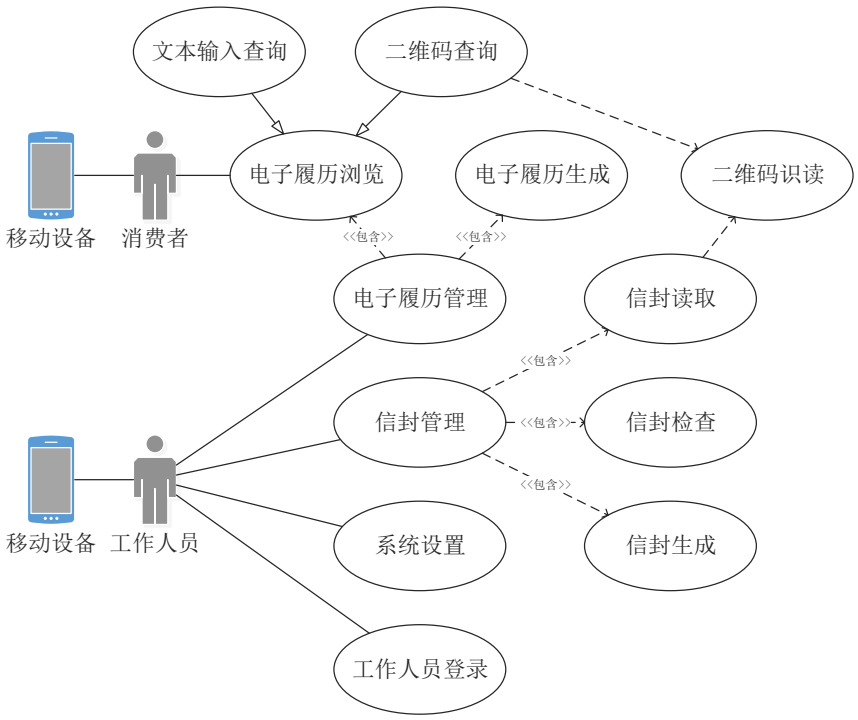

图 3.1 系统用例图

3.1.3 用例需求分析

3.1.3.1 系统设置

当工作人员第一次打开本移动应用系统准备使用时，需要先进行初始化设置，以连接所属公司的子服务器，并且在之后也可以对设置进行修改。具体步骤为：工作人员点击“服务器设置”，移动端跳转至设置界面；工作人员填写子服务器的 IP 地址后点击连接，移动端对 IP 进行连接，进行确认，如果 IP 地址填写错误，导致连接失败，提示填写错误；如果 IP 正确，子服务器连接成功并认证后返回公司信息。移动端提示连接成功，并显示公司信息。

3.1.3.2 工作人员登录

当移动端已成功与子服务器连接且工作人员在该子服务器上已成功注册之后，工作人员可以进行登录操作，具体步骤为：工作人员点开登录页面，在登录页面输入用户名和密码；移动端将用户名和密码发送给连接的子服务器，请求进

行身份验证;子服务器返回身份验证信息,若认证成功,返回工作人员用户信息;若失败，提示错误，若认证成功，跳转至工作人员界面。

3.1.3.3 电子履历浏览

当移动端已成功与子服务器连接并且用户持有产品序列号,则可以通过本移动应用系统查询该产品的对应电子履历并进行浏览。对于消费者来说，可以在购买产品之后通过包装上的二维码或者序列号进行查询，对于工作人员来说，可以在生产线上对产品进行查询。具体步骤为：用户点击“查询履历”，系统显示履历查询页面；用户选择扫描二维码或者手动输入商品序列号。移动端发送产品序列号至连接的子服务器，请求相应的履历文件。若产品履历文件不存在，则提示履历文件不存在，若存在，则子服务器返回查询到的履历文件；移动端将接收到的履历文件存储在本地缓存中，同时对履历文件进行解析，然后以倒叙的方式，将文件所包含的信息直观地显示在界面上。

3.1.3.4 电子履历生成

当位于生产线的工作人员成功登录并获取产品序列号,可以根据不同的事件类型和履历生成信息，对产品生成对应的履历文件，具体步骤为：处于生产线的工作人员，通过扫描二维码得到产品序列号；根据产品序列号，移动端向子服务器端请求事件信息。子服务器端在事件盒中进行搜索。若搜索成功，则返回给移动端，移动端显示事件信息，且该信息可被编辑；如果事件盒中没有信息即搜索失败，则移动端提供编辑界面让工作人员自行选择阶段（事件类型），并手动输入履历生成所需的信息；工作人员确认履历生成所需信息后，选择“生成履历”。根据事件类型的不同，若为出生履历类型，则不需要先期履历文件，直接生成履历；否则移动端根据产品序列号，向子服务器端请求先期电子履历文件，子服务器端搜索并返回对应的先期履历文件,移动端接收后将先期履历文件保存至本地缓存；移动端根据事件类型、履历生成信息和先期履历文件生成新的电子履历文件，并进行数字签名；工作人员预览无误后选择新履历上传，子服务器将新履历上传至中央服务器，并返回上传成功。移动端更新本地数据库。

3.1.3.5 二维码识读

产品序列号或者信封 ID 以二维码标签作为数据载体,依附于产品包装之上,伴随产品经过供应链的各个环节。当消费者或工作人员在产品包装上获取二维码后，需要通过二维码识读获取其中的信息，以便进行下一步操作，具体步骤为：用户点击“二维码扫描”，系统打开手机摄像头，进入扫描界面；用户将手机摄像头对准二维码进行扫描，系统识读二维码信息，获得商品序列号或者信封 ID 并

显示在界面上，然后自动跳转至下一界面。

3.1.3.6 信封读取

当工作人员成功登录且产品批次附有信封二维码时，工作人员可以读取信封内所包含的所有产品的履历信息，具体步骤为：用户扫描信封的二维码，移动端识别二维码并获取信封 ID；移动端将信封 ID 发送至连接的子服务器，请求相应的信封文件。子服务器返回查询到的信封文件；移动端将接收到的信封文件保存至本地缓存，并对信封文件进行解析，获取信封内所有产品的序列号，并将信封内所有产品的信息直观地显示在界面上；工作人员可以点击对应的产品，查看其具体的履历信息。

3.1.3.7 信封检查

当工作人员成功读取信封后，可以逐一扫描该批次的所有产品的二维码进行信封检查，以检查是否有产品遗漏或出错，具体步骤为：工作人员选择信封检查，进入产品扫描界面；工作人员逐一扫描到达批次所有产品的二维码，与信封内的产品序列号进行比对，检查成功的产品显示已检查；工作人员扫描完所有产品后，点击“扫描结束”，移动端界面显示最终的检查结果。

3.1.3.8 信封生成

当某一批次产品到达后，工作人员可以在登录后逐一扫描该批次产品，并生成对应的信封，具体步骤为：工作人员选择信封生成，进入产品扫描界面；工作人员逐一扫描待发货批次所有产品的二维码；工作人员扫描完所有产品后，点击“扫描结束”，移动端显示扫描的所有产品信息及产品总数；工作人员确认无误后，输入信封生成所需信息，选择生成信封；移动端根据所有产品序列号和信封生成信息生成新的信封文件，上传至子服务器端，上传成功后更新本地数据库；工作人员选择打印包含新的信封文件 ID 信息的二维码。

3.2 业务流程分析

本系统需要对产品在整个供应链过程中的各个环节进行分析，从而针对整个业务流程对移动电子履历系统进行设计，不同的电子履历类型对应不同的业务流程环节，并且涵盖了该环节中所涉及到的数据，从而使得本系统更加具有实际应用价值，更好地适用于复杂繁琐的产品业务流程，记录更为全面完整的流程数据。通过将移动端应用到实际中，提供更为方便快捷的业务流程操作。

3.2.1 产品供应链

在供应链流程中，产品从生产到流通到最后销售会经过多个环节。一般来说，

一件产品首先由生产商进行生产，之后可能会经历检验检疫、加工、重新包装等环节[31]，并且有可能这些环节不是在同一家公司进行的，则需要通过“发货——运输——收货”的过程将产品由公司 A 传递至下一公司 B。最后产品会投放到市场上，供消费者购买。

生产环节是指产品最初被生产商生产出来，或者被制造商制造出来，或者出生的流程，需要记录最原始的生产、制造或者出生过程中的相关信息以及产品的基本信息，生成相应的初始产品履历文件。

检验检疫环节是指产品被检验检疫人员进行质量检查的流程，需要记录产品的质量信息和检查结果等信息，生成相应的检验检疫履历。

加工环节是指产品被加工人员进行进一步地加工操作的流程，需要记录加工过程中的相关操作信息和环境信息等，生成相应的加工履历。

重新包装环节是指产品被包装人员进行重新组合包装的流程，需要记录相关操作信息等，生成相应的重新包装履历。

发货环节是指产品被一个公司（发送方）发往下一个公司（接收方）的流程，需要记录发货时的相关信息，生成相应的发货履历。

运输环节是指产品被发出后通过运输前往目的地的流程，需要记录运输过程和路途中的相关信息，生成相应的运输履历。

收货环节是指产品到达目的地，被接收方接收的流程，需要记录收货时的相关信息，生成相应的收货履历。

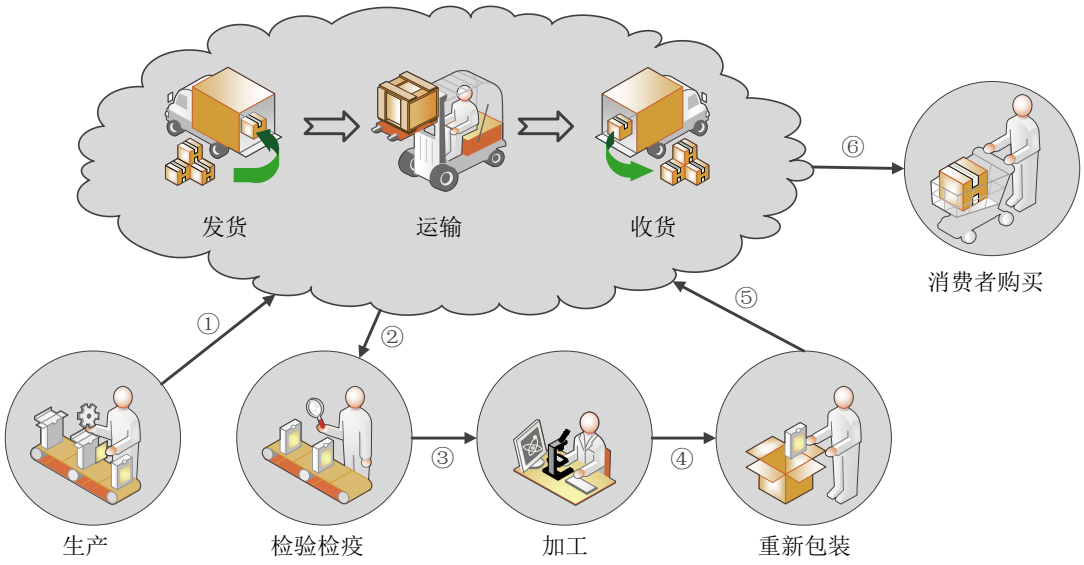

图 3.2 产品供应链业务流程示例图

如图 3.2 所示，展示了一件产品所经历的整个流程的例子。产品在公司 A 被生产出来后，经过发货、运输到达下一公司 B，公司 B 进行收货后，对其进行检验检疫、加工、重新包装等一系列操作，再通过发货、运输到达下一公司 C，公司 C 收货后将产品上架，供消费者购买。

3.2.2 移动端实际应用场景

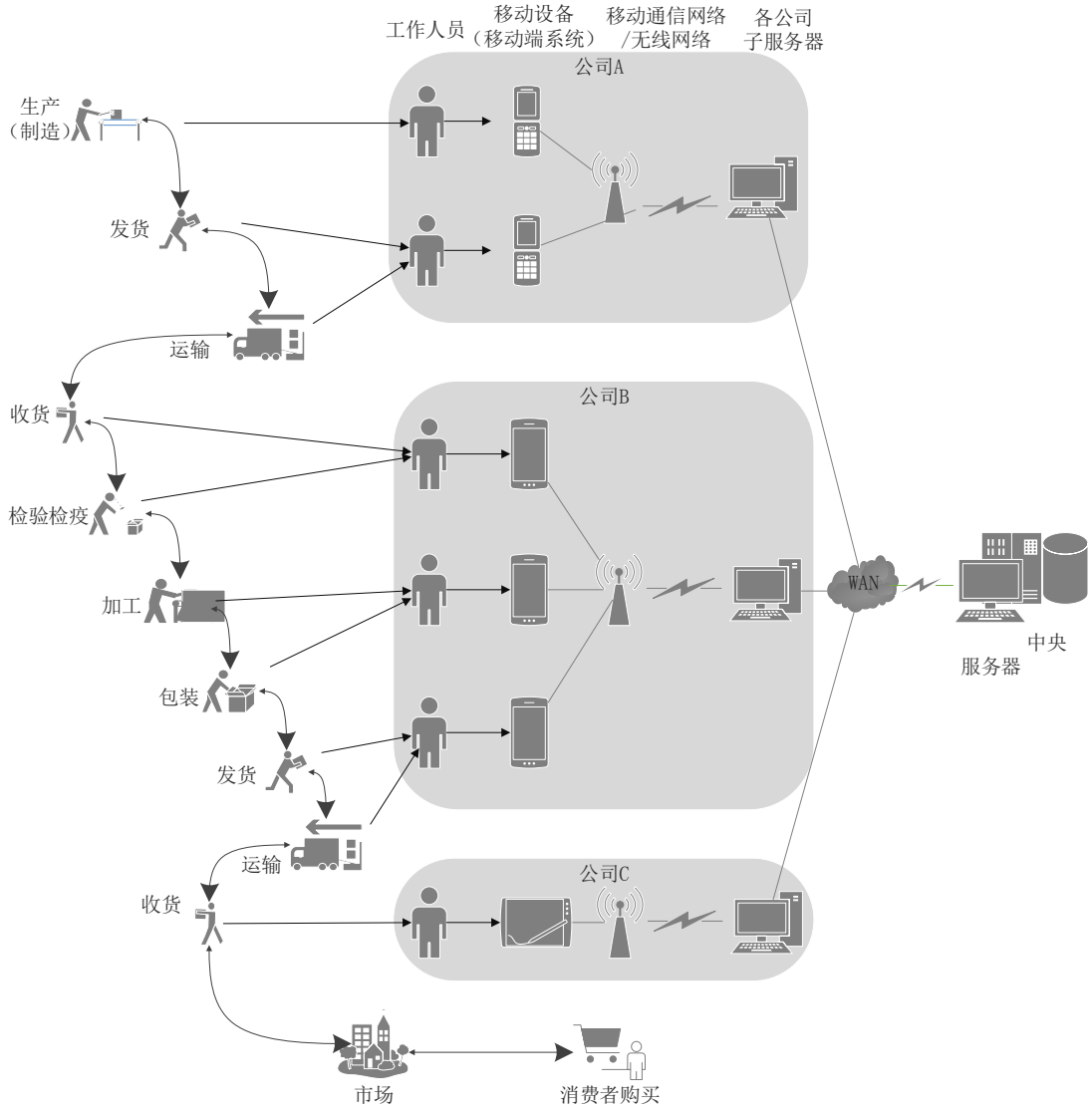

图 3.3 移动电子履历系统实际应用场景

本论文的移动电子履历系统需要根据产品供应链的业务流程的特性进行设计和实现，考虑到业务流程繁琐，生产线环境复杂，为了使得位于生产线的工作人员能够便携地使用电子履历系统，不受环境的约束，本论文采用移动端技术，为每个工作人员提供一个手持移动设备，通过该设备上的移动电子履历系统，在任何位置、任何场景，简单方便快捷地完成电子履历系统的各个业务流程。工作人员可以通过移动端，在每个业务环节中，比如生产、加工、收货时，进行相应的履历查看或者履历生成等操作，并且通过移动端与各自公司对应的子服务器进行数据通信。移动端利用移动通信网络或者无线网络，将生产线上的数据和信息通过电子履历文件的形式发送给子服务器端，子服务器端再通过网络将信息发送给中央服务器进行汇总存储，同时，移动端还可以向子服务器端请求其存储的数据，比如先期履历文件，子服务器端通过向中央服务器的数据库查询后返回给移

动端。

如图 3.3 所示，展示了移动电子履历系统在产品供应链业务流程中的实际应用场景流程，是根据图 3.2 的业务流程实例设计的。在该实例中，展示了一件产品经历的整个流程，产品首先在公司 A 被生产出来，再经过发货、运输环节，对应每个环节，公司 A 的生产线上的工作人员都会通过移动端系统生成对应的履历文件，发送给公司 A 的子服务器 a，子服务器 a 再发送给中央服务器；公司 B 接收到产品后，经过检验免疫、加工、重新包装、发货、运输环节，对应每个环节，公司 B 的生产线上的工作人员都会通过移动端系统生成对应的履历文件，发送给公司 B 的子服务器 b，子服务器 b 再发送给中央服务器；公司 C 接收到产品后，以同样的方式生成收货履历，然后将产品发往市场，在商场、超市、零售店等上架，供消费者挑选购买。

3.2.3 面向移动的分析

3.2.3.1 感知能力

移动设备具有多种传感器，比如：摄像头、麦克风、GPS（全球定位系统，Global Positioning System）、NFC（近场通信，Near Field Communication），提供了多种感知能力。对于本系统来说，可以通过键盘手动输入序列号，也可以使用二维码标签作为数据载体，通过摄像头扫描二维码，读取识别出其中包含的信息。利用移动设备的 NFC 感知能力，还可以使用 NFC 标签存储数据，通过设备自带的 NFC 技术读取或写入数据，但考虑到目前具有 NFC 功能的移动设备尚占少数，不具有普遍性和普及率，故将在未来的工作中对其进行实现。

3.2.3.2 移动性、便携性和易用性

移动设备本身具备移动性和便携性。用户可以很方便地手持移动设备，移动至在不同的位置使用该设备。同时移动设备一般是智能手机或者平板，相对笔记本电脑、台式机等设备来说极为轻便，易于携带。从而使得位于流水线上的工作人员，可以手持手机或者平板移动到各个位置，方便快捷地对位于各个位置的产品进行扫描、检查、生成履历等操作，从而提高工作效率，简化流程。

由于手机或平板屏幕的大小限制，本系统的应用界面要设计得简洁、人性化、易理解，使用户一目了然，并且保证操作流程简单、方便、高效，符合用户的操作习惯，提高交互性，从而保证本系统的易用性。

3.2.3.3 安全性

由于 Android 应用程序没有提供完整成熟的安全机制，Android 平台存在一些安全威胁和漏洞，但电子履历系统要求保障履历信息和相关数据的机密性、完

整性和可信任性，故本论文需要采取一些措施提供移动电子履历系统的安全性。本系统需要采取数字签名技术对电子履历进行签名认证，从而保证履历文件的完整性，提高履历的可信任性。同时，通过 Android 平台自带的加密算法对传输和存储的数据进行加密，以提供数据的安全保护。

第四章 移动电子履历系统的设计

4.1 系统总体框架

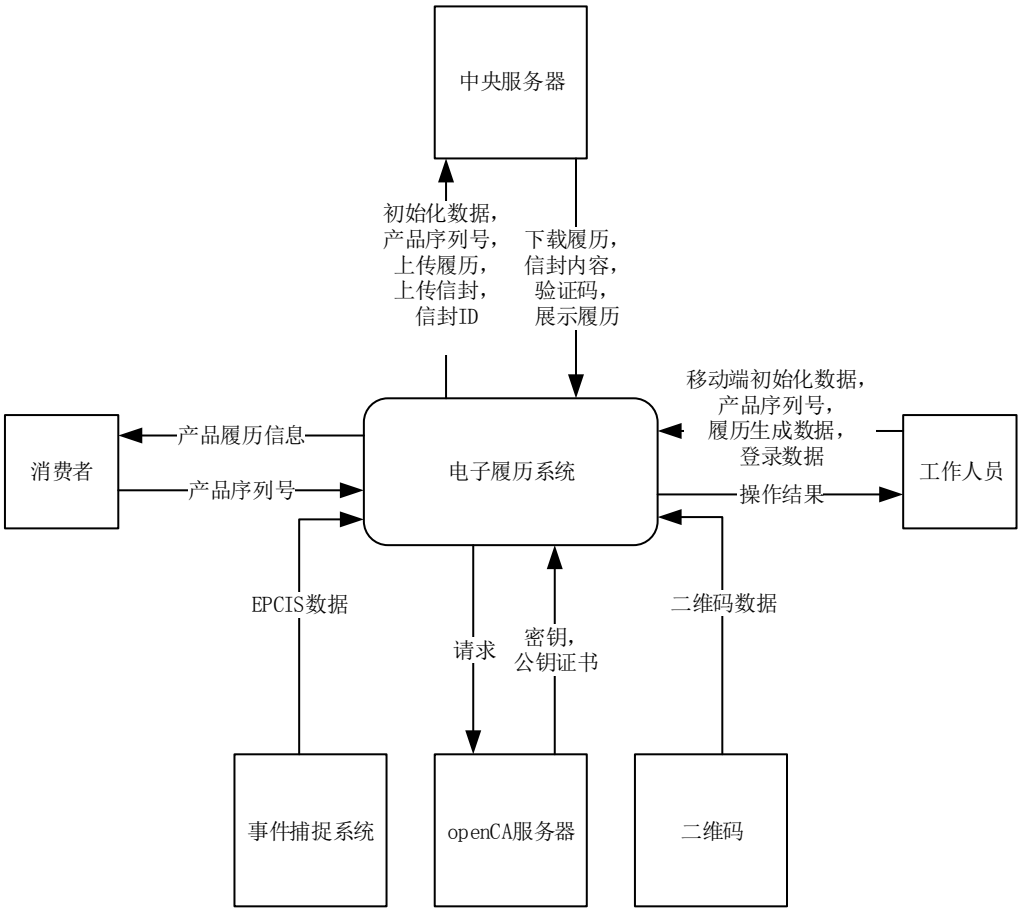

图 4.1 系统顶层数据流图

如图 4.1 所示，该数据流图描述了电子履历系统与周围环境的数据交互情况。周围环境包括：参与角色（消费者和工作人员），外部系统（中央服务器系统、事件捕捉系统、OpenCA 服务器）以及数据载体（二维码）。其中，中央服务器与电子履历系统进行数据交互，用来实现部署于农产品生产线上的履历系统的初始化注册，以及电子履历文件和信封文件的上传和下载；消费者与电子履历系统进行交互，用来实现产品电子履历信息的查询和浏览；工作人员与电子履历系统进行交互，用来实现电子履历子服务端和移动端的初始化设置、履历的生成和查询、信封的浏览、检查和生成等功能；OpenCA 是一个开源的证书授权中心（Certificate Authority），OpenCA 服务器与电子履历系统进行交互，实现了密钥管理模块的服务托管；事件捕捉系统负责将生产线上产生的产品数据高效准确地上传给电子履历系统；电子履历系统通过二维码标签读取其载有产品序列号等信息。

如图 4.2 所示，该数据流图描述了电子履历系统两个子系统之间的数据交互情况。对于周围环境来说，电子履历子服务器系统主要负责和外部系统（中央服

务器系统、事件捕捉系统、OpenCA 服务器）以及管理员进行交互，电子履历移动端主要负责和工作人员以及二维码进行交互。同时，子服务器与移动端之间提供了多个接口进行多种数据交互，用来实现移动端的初始化设置、工作人员的登录、电子履历文件和信封文件的查询、生成、上传和下载等功能。

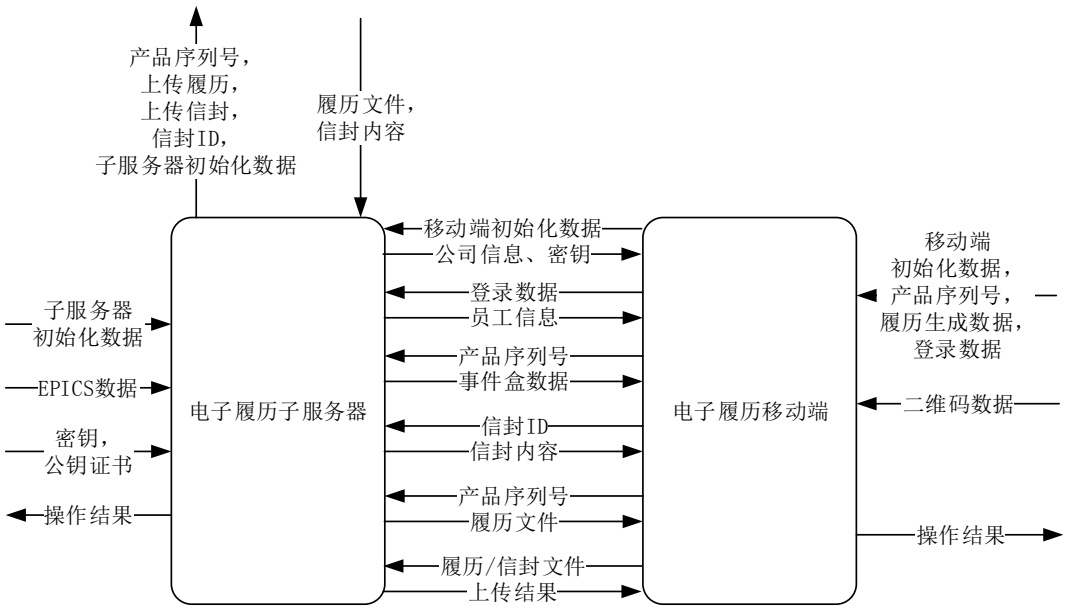

图 4.2 电子履历系统数据流图

电子履历移动端不会与中央服务器进行直接的数据交互，对于履历和信封的上传和下载都是通过子服务器系统间接地实现，从而保证数据的安全隔离。

具体来说，所有的电子履历文件和信封文件都存储在中央服务器上，而每个子服务器上只存储有本地的电子履历文件和信封文件。

对于文件的下载来说，若移动端需要根据产品序列号获取文件，则移动端首先搜索本地数据库，若未找到该文件，则向子服务器发起文件下载的请求。子服务器在其本地数据库中进行检索，若找到则将该文件的数据流返回给移动端，若没有找到则由子服务器向中央服务器发起文件下载请求。中央服务器将查找到的文件数据流返回给子服务器端，子服务器再将该数据流返回给移动端，同时将其存储在本地并更新数据库以方便之后的使用。

对于文件的上传来说，若移动端需要上传本地生成的新的文件，则移动端首先将文件数据流发送给子服务器，子服务器将其转发给中央服务器，中央服务器将接收到的数据流成功存储并成功更新数据库，然后向子服务器发送操作成功的消息。子服务器收到消息后将数据流存储在本地并更新数据库，然后向移动端发送操作成功的消息，移动端接收消息后对本地数据库进行更新。

如图 4.3 所示，该数据流图描述了电子履历移动端系统的主体框架。该系统主要由初始化设置、登录、二维码扫描、履历管理、信封管理几个功能模块组成。

电子履历移动端系统作为 Android 应用安装在移动设备上，极大地方便工作人员的使用。位于产品生产线上的每个工作人员都可以手持一台移动设备，比如：手机、平板，安装该 Android 应用便可进行业务操作。

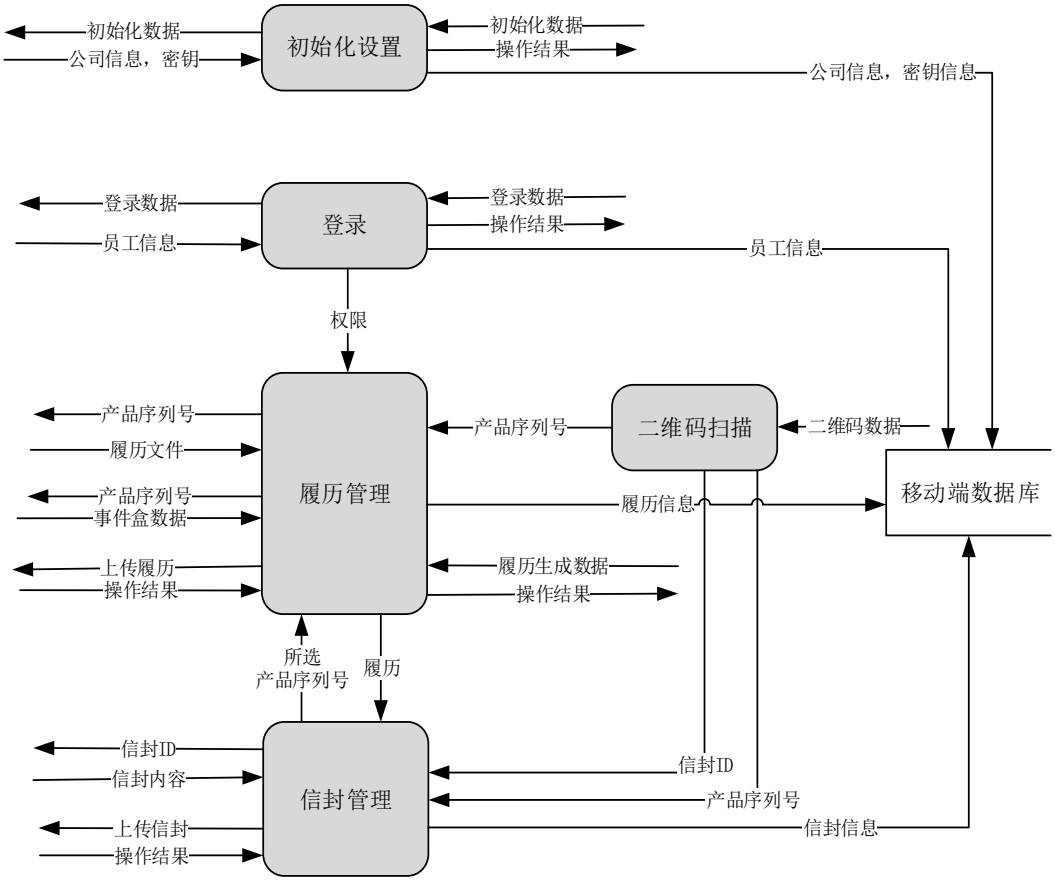

图 4.3 电子履历移动端数据流图

安装完成后初次使用时，需要由工作人员对该应用进行初始化设置，输入所属公司对应子服务器的 IP 等信息，确认成功连接以保证之后各模块与子服务器之间的通信。并向子服务器发起请求，获得公司具体信息和私钥、公钥证书等数据并存储在本地数据库，用以履历管理模块的生成功能。

初始化设置后，工作人员需要通过登录模块进行身份验证。移动端将工作人员输入的用户名和密码发送给子服务器端进行验证，返回验证结果。只有验证成功才能获得履历管理模块和信封管理模块的操作权限。

二维码扫描模块通过手机摄像头对产品上的二维码标签进行扫描，经过解码后获得产品序列号、信封 ID、文件类型等信息，用来提供给履历管理模块和信封管理模块。

履历管理模块由二维码扫描模块获取产品序列号信息，由子服务器获得事件盒数据和先期履历数据，由本地数据库获得公司信息和密钥，由生产线上的工作人员获得履历生成数据，生成相应的履历文件并进行签名、上传。每一个产品都

会对应一个电子履历文件，在产品经历生产、运输、销售等流程时，其对应的履历也会层层嵌套不断扩展，随之在生产线上流动。用户还可以通过输入产品的序列号或者扫描二维码对履历进行查询，移动端会将查询到的履历内容以倒叙的方式，直观地呈现在设备屏幕上。

信封管理模块由二维码扫描模块获取信封 ID 信息和产品序列号，由子服务器获得信封内容，从而对信封进行查看、检查和生成，还可以通过履历管理模块浏览信封中相应产品的履历。在生产线的每个结点中，工作人员在进行收货时，会根据上一个结点随产品发来的信封，检查收到的产品是否有误，并在发货时，将要发出的产品打包生成新的信封，随这一批产品发往下一个结点。

4.1.1 履历管理模块

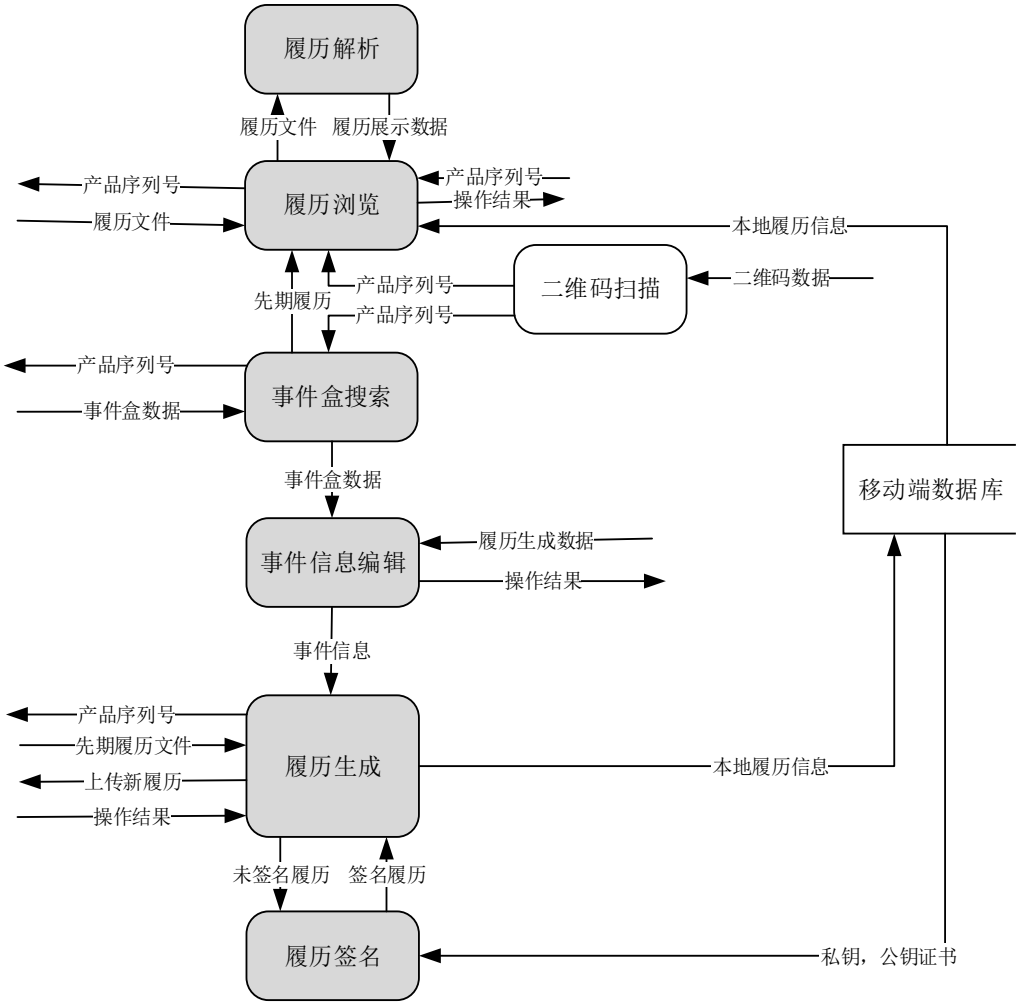

图 4.4 履历管理模块数据流图

如图 4.4 所示，该数据流图为移动电子履历系统的履历管理模块的主体框架。履历管理模块主要实现了履历信息浏览和履历文件生成两大功能。重要的功能模块包括：履历解析、履历浏览、事件盒搜索、事件信息编辑、履历生成、履历签名，各模块之间进行数据交互，结合由二维码扫描模块和移动端数据库获取的数

据，协同合作实现履历管理。

4.1.1.1 履历信息浏览

用户可以通过移动设备的键盘文本输入产品序列号，或者通过摄像头扫描产品上附着的二维码标签，二维码扫描模块会对图像进行解析并将解析出的序列号传给履历浏览模块。

履历浏览模块首先会访问移动端本地数据库，根据产品序列号进行检索，若检索成功则取得对应的履历文件，若检索失败则将产品序列号发送给子服务器以请求履历下载，成功下载对应的履历文件。

接着，履历浏览模块将履历文件传给履历解析模块。履历解析模块根据电子履历标准，从该 XML 文件中读取产品的整个生产线信息，并将其按照时间的顺序储存在定义好的结构体中返回。

履历浏览模块再对结构体中存储的履历展示数据进行处理，将产品履历中的各个生产线流程，以倒叙的方式，详细、直观地展示在履历浏览界面上。

4.1.1.2 履历文件生成

位于生产线上的工作人员通过摄像头扫描产品上附着的二维码标签，二维码扫描模块会对图像进行解析以获取产品序列号。

事件盒搜索模块将产品序列号发送给子服务器以请求查询事件盒数据。若子服务器中已存在该产品的事件盒数据，则将搜索到的数据返回。

事件盒搜索模块将收到的事件盒数据传递给事件信息编辑模块，事件信息编辑模块将其显示在编辑界面上。工作人员可以通过编辑界面输入履历生成所需数据，对事件盒数据进行编辑，完善事件信息。若事件盒数据为空或者不完整，则空缺的数据部分由工作人员填补。编辑完成确认无误后将完整的事件信息传给履历生成模块。

履历生成模块通过产品序列号检索本地履历数据库，若本地不存在，则向子服务器发送履历下载请求，以获得对应的先期履历文件。同时，还可以通过履历浏览模块查看先期履历的展示数据。然后结合事件信息和先期履历文件，根据不同的事件类型，生成对应类型的新履历文件。若履历需要签名，则将未签名的履历传给履历签名模块。

履历签名模块访问本地数据库，获得私钥和公钥证书，对履历文件进行数字签名，签名成功后将签名履历返回给履历生成模块。

履历生成模块将成功生成并签名的履历文件上传给子服务器，若上传成功，则将其保存在本地，并更新本地数据库，将新的履历信息写入本地的履历数据库中。

4.1.2 信封管理模块

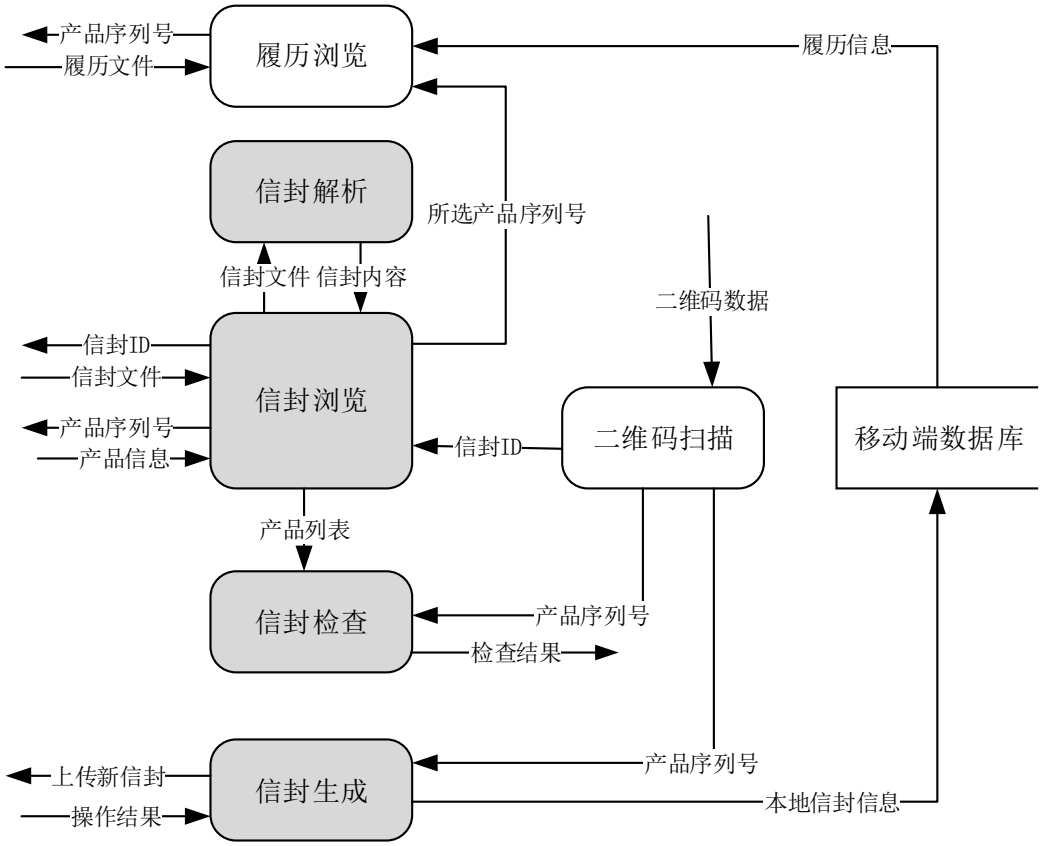

图 4.5 信封管理模块数据流图

如图 4.5 所示，该数据流图为移动电子履历系统的信封管理模块的主体框架。履历管理模块主要实现了信封浏览、信封内容检查、信封文件生成三大功能。重要的功能模块包括：信封搜索、信封内容列表、信封检查、信封生成，各模块之间进行数据交互，联合二维码扫描模块和履历浏览模块，结合由移动端数据库获取的数据，协同合作实现信封管理。

4.1.2.1 信封浏览

位于生产线上的工作人员通过摄像头扫描某批产品上附着的二维码标签，二维码扫描模块会对图像进行解析并将解析出的信封 ID 传给信封搜索模块。

信封浏览模块首先会访问移动端本地数据库，根据信封进行检索，若检索成功则取得对应的信封文件，若检索失败则将信封 ID 发送给子服务器以请求下载信封，成功下载对应的信封文件。

接着，信封浏览模块将信封文件传给履历解析模块。信封解析模块根据电子履历标准，从该 XML 文件中读取该信封的生产线数据以及所包含的所有产品的序列号列表，并储存在定义好的结构体中返回。

信封浏览模块再对结构体中存储的信封内容进行处理，将产品序列号列表中

的所有序列号发送给子服务器，以请求所有的产品信息（包括履历 ID、产品名称、可信度等）。然后将信封生产线数据以及返回的所有产品信息以列表的形式，详细、直观地展示在履历浏览界面上。同时，工作人员可以选中某个产品，通过履历浏览模块，查看其履历的详细内容。

4.1.2.2 信封内容检查

接收到某批产品时，工作人员可以对信封内的产品列表进行检查。信封检查模块通过信封浏览模块获取某个信封的产品列表，并将产品列表中每一个产品的检查状态初始化设置为未检查。工作人员通过摄像头依次扫描该批产品中每一个产品上附着的二维码标签，二维码扫描模块会对图像进行解析并将解析出的产品序列号传给信封检查模块。

信封检查模块依次根据序列号在产品列表中进行查找，若查找成功，将产品列表中对应产品的检查状态设置为检查，若查找失败，将产品列表中对应产品的检查状态设置为错误。所有产品都扫描完成后，展示给工作人员检查结果，若有错误，则交付人工处理。

4.1.2.3 信封文件生成

发出某批产品时，工作人员可以对这批产品生成相应的信封。工作人员通过摄像头依次扫描该批产品中每一个产品上附着的二维码标签，二维码扫描模块会对图像进行解析并将解析出的产品序列号传给信封生成模块。

工作人员输入一些生产线信息后，信封生成模块根据产品序列号列表和生产线信息，按照信封电子履历标准，生成对应的信封文件，并将其上传给子服务器，若上传成功，则将其保存在本地，并更新本地数据库，将新的信封信息写入信封数据库中，并将新的信封 ID 生成二维码标签，附在该批产品上随之发往下一个结点。

4.2 数据模型

4.2.1 履历模型

4.2.1.1 履历种类

电子履历涵盖了农产品的销售流程，以及每一个销售流程中所涉及到的数据。履历的种类是根据农产品生产流程进行定义的，主要包含以下几种：

1) 初始环境履历：记录农产品最初种植或养殖的环境信息。
2) 环境履历：记录农产品种植养殖过程中周围动态变化环境因素的电子履历。内部通常嵌套一份初始环境履历或者另一份环境履历。

3) 出生履历：记录农产品的出生信息。可以被初始产品履历嵌套。
4) 初始产品履历：记录农产品正式包装成为产品的过程。必要时，会嵌套相关的出生履历和环境履历。
5) 发货履历：记录农产品从一个厂商运输到另一个厂商时的发货信息。
6) 收货履历：记录当厂商接收到农产品时生成的收货信息。
7) 未签署的收货履历：主要属性和收货履历类似，但未进行数字签名。
8) 重新包装履历：记录农产品被工厂重新组合包装的相关信息。
9) 加工履历：记录农产品在加工过程中一些信息的履历。
10) 运输履历：记录农产品在运输路途中的信息。
11) 未签署的运输履历：主要属性和运输履历类似，但未进行数字签名。
12) 检验检疫履历：记录农产品被送往检测中心或者在其他过程中的检测信息。

4.2.1.2 履历元素数据模型

电子履历元素遵循 XML Schema 规范。每个 XML 元素都采用一种类型（Type）来表示，例如：电子履历 pedigree 将表示成 LayerType，发货履历 shippedPedigree 将表示成 shippedPedigreeType。

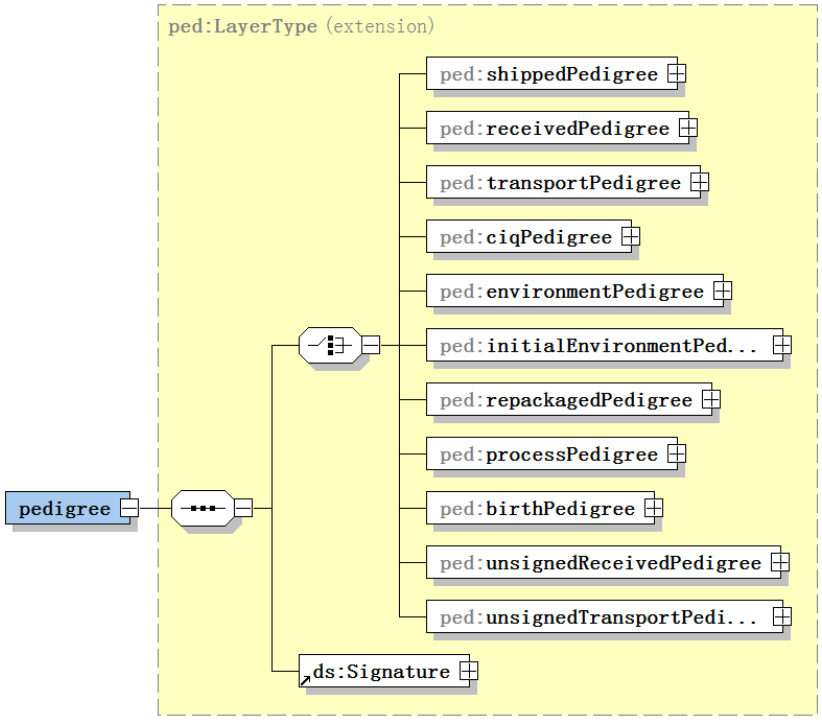

图 4.6 电子履历数据模型

图 4.6 描述了电子履历元素所应遵循的 XML 标准。pedigree 元素使用<LayerType>来表示，代表了产品生产流程中，生产线上所传递的履历文件。该元素由一份事件履历和数字签名组成。其中，本农产品电子履历系统支持 11 种

事件履历，分别对应农产品生产流程中的相关事件，包括：发货履历、收货履历、运输履历、检验检疫履历、环境监测履历、初始环境履历、重新封装履历、重新加工履历、出生履历以及未签署的收货履历和未签署的运输履历。数字签名元素保障了该份电子履历文件的合法性和完整性。如表 4.1 所示，展示了<LayerType>所包含的所有元素。

表 4.1 电子履历元素详细说明

名称	类型	描述
shippedPedigree	shippedPedigreeType	发货电子履历：记载了产品在生产供应链之间进行传递时的发货信息。通常在产品发货时产生。
receivedPedigree	ReceivedPedigreeType	收货电子履历：记载了产品在生产供应链之间进行传递时的收货信息。通常在产品收货时产生。
transportPedigree	TransportPedigreeType	运输电子履历：记录了产品在生产供应链之间进行传递时的运输信息。通常在产品运输过程中产生。
ciqPedigree	CiqPedigreeType	检验检疫电子履历：记录了产品在相关检测部分进行检验检疫的信息。通常在检验检疫后产生。
environment-Pedigree	Environment-PedigreeType	环境监测电子履历：记录生产产品的环境的相关数据。通常在每次环境监测后产生。
initialEnvironment-Pedigree	InitialEnvironment-PedigreeType	初始环境电子履历：记录产品生产环境在初次建立时的相关数据。通常在环境建立时产生。
repackagePedigree	Repackage-PedigreeType	重新包装电子履历：记载了产品在供应商处进行重新包装的相关信息。通常在产品重新包装后产生。
processPedigree	ProcessPedigreeType	重新加工电子履历：记载了产品在生产商处进行重新加工的相关信息。通常在产品重新加工后产生。
birthPedigree	BirthPedigreeType	出生电子履历：记载了产品在出生或者收货时的相关基本信息。在产品出生或收货时生成。
Signature	ds:signatureType	数字签名：此属性包含了和数字签名相关的基本信息，如：签名算法，证书属性，证书发行者的属性，签名值。此元素根据上述事件履历的内容生成。

4.2.1.3 履历文件结构

电子履历在本系统的具体表现形式为符合 EPCglobal 标准的 XML 文件。履历文件是随着生产流程层层嵌套的。

履历文件最深处的元素一般是一个 initalPedigree 或者是一个 repackagedPedigree 元素。当制造商或者批发商为产品创建一个新的履历时，一

般使用 intialPedigree 元素。而重新包装或装配的产品一般是使用 repackagedPedigree 元素。产品经历每个事件，都会产生对应事件类型的新的履历元素，比如收货时会生成 receivedPedigree，并且生成的新的履历元素会被 pedigree 层元素包裹其中。这些连续的层代表了产品经历的整个产业链中的各个流程及相应信息。

4.2.2 信封模型

信封是提供给用户的一种可选方案。信封是一种在传输过程中，由发送方发往接收方的产品集合的电子包装。其包含的产品序列号信息可以用来在接收时检查产品的数量等是否存在问题，并能方便由产品到履历的匹配，可以用来批量地生产收货履历。

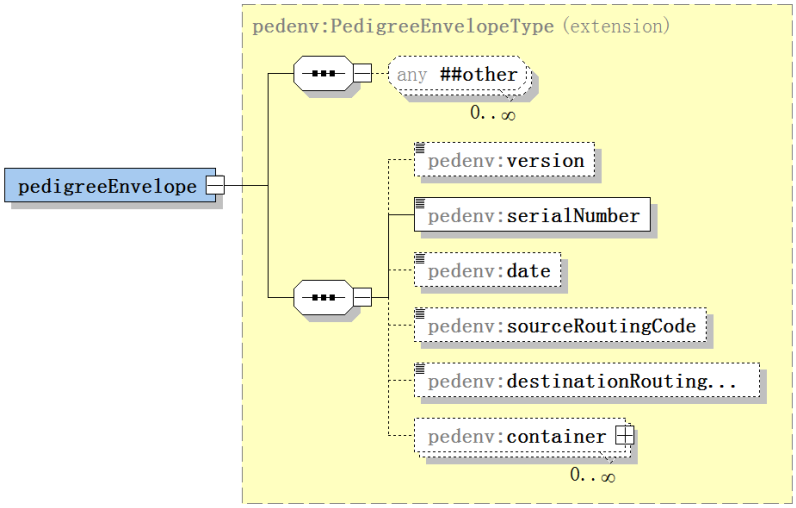

图 4.7 信封数据模型

如图 4.7 所示，信封遵循 XML Schema 标准。pedigreeEnvelope 元素使用 PedigreeEnvelopeType 表示，代表产品集合在发货时形成的信封文件。信封一般在发货时生成，工作人员将发往同一接收方的同批次产品划分为同一产品集合，该元素需要以一定的标准形式储存日期、来源地代码、目的地代码和产品集合内所有产品的序列号等信息。该元素可以用来在收货时检查是否有产品丢失、差错等问题，方便工作人员对产品进行管理，还可以用于批量生成收货履历。如表 4.2 所示，描述了信封所包含元素的详细说明。

表 4.2 信封所包含元素详细说明

名称	类型	强制性	描述
version	xs:string	是	版本号：此信封的当前版本编号。
serialNumber	xs:string	是	产品序列号列表：信封内所有产品的序列号。
date	xs:dateTime	否	日期：此信封生成并传输的日期。
sourceRoutingCode	xs:string	否	来源地代码：发送方的参考或方位代码。
destinationRoutingCode	xs:string	否	目的地代码：接收方的参考或方位代码。

4.3 关键流程分析

4.3.1 履历生成流程

履历生成模块是本系统最重要的业务功能模块，具体过程是：位于生产线的工作人员手持安装本系统 Android 应用程序的移动设备，扫描产品上附着的包含产品序列号的二维码标签，根据不同的事件类型（履历类型）和履历生成信息，生成相应的签名或未签名的履历文件，然后成功上传至子服务器，同时更新本地数据库。

本系统提供了不同类型的履历，根据事件类型的不同，分别对应产品生产流程中的相关事件。根据不同的履历类型，履历生成的流程有所区别，以下三个履历生成模块分别描述了出生履历、收货履历和其他一般履历在生成过程中的具体流程和操作。

4.3.1.1 出生履历生成流程

出生履历记录了产品的出生信息，可以被初始履历所嵌套。出生履历的属性中不包括其他电子履历，即出生履历不会嵌套其他履历，故在出生履历生成过程中，不需要进行先期履历的查找、下载和嵌套，直接根据出生信息生成对应的履历文件即可。

具体流程如图 4.8 所示：

步骤 1：工作人员在移动端打开履历生成界面，扫描包含产品序列号的二维码；移动端将序列号发送给子服务器，子服务器搜索其本地事件盒，返回对应的事件数据或未找到；

步骤 2：移动端接收到返回的信息，显示出生事件编辑界面，如果返回信息中有事件数据，则将数据显示到编辑界面上对应位置，如果没有事件数据，则显示空白。工作人员可以通过编辑界面，手动输入出生事件信息进行完善，最后选择确认生成，若取消则跳回步骤 1 重新扫描。

步骤 3：移动端生成出生履历文件，并显示新履历的预览界面。

步骤 4：若工作人员选择取消上传，则跳回步骤 2 重新编辑事件信息；若确定上传，则移动端将新的履历文件发送给子服务器，子服务器再将履历文件发送给中央服务器。若中央服务器存储成功，则返回保存成功的消息给子服务器。若子服务器存储成功，则返回消息给移动端。移动端接收到保存成功的消息后更新本地数据库。

4.3.1.2 收货履历生成流程

收货履历记录当厂商接收到农产品时生成的收货信息。由于本系统可以通过

信封打包产品，同一信封内的产品的收货信息（如收货时间等）是相同的，故可以工作人员在收到某一批产品时，可以对其信封内的所有产品批量生成收货履历，且同时完成对信封内产品的检查。

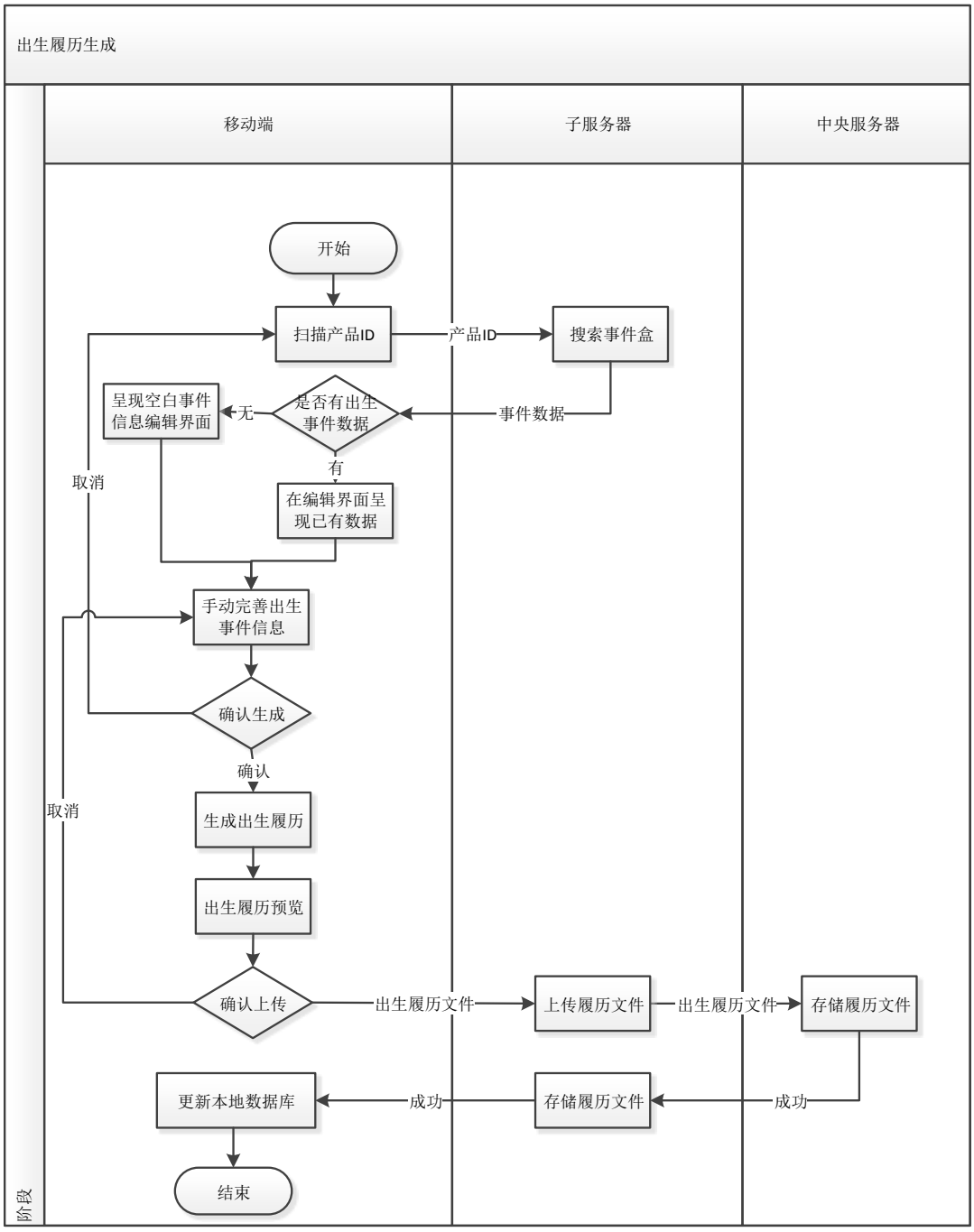

图 4.8 出生履历生成流程图

具体流程如图 4.9 所示：

步骤 1：工作人员在移动端打开履历生成界面，选择信封收货，扫描包含信封 ID 的二维码；

步骤 2：移动端将信封 ID 发送给子服务器，子服务器搜索其本地数据库，查找信封文件及其所包含的产品的先期履历文件，若未查找到，则子服务器向中

央服务器请求，中央服务器搜索成功后返回文件，子服务器再将信封内所有产品序列号及先期履历返回给移动端；

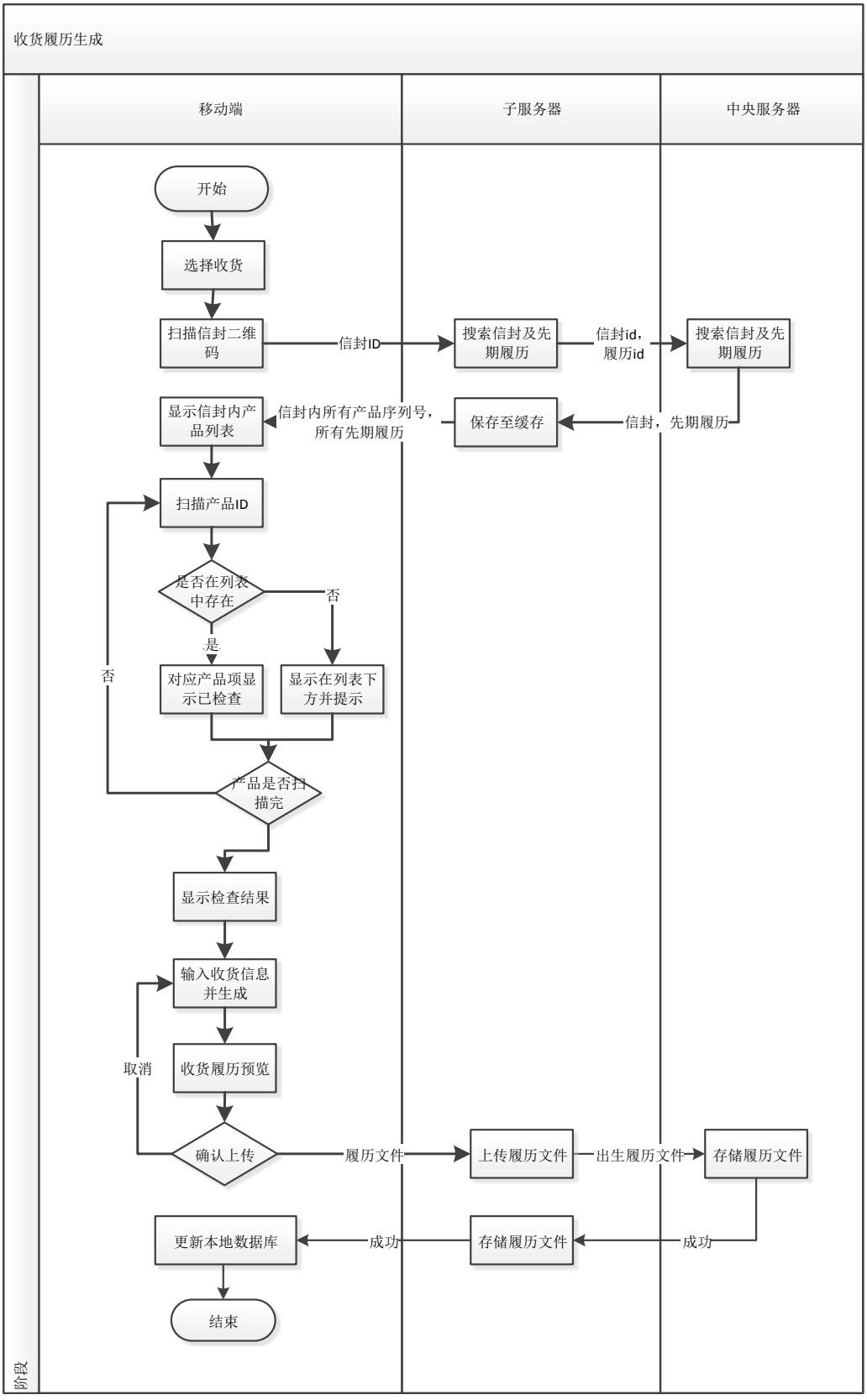

图 4.9 收货履历生成流程图

步骤 3：移动端将所有先期履历保存至缓存，并将信封内产品序列号列表显示在界面上。

步骤 4：工作人员开始依次扫描收到的产品，如果该产品在列表中存在，则列表中该产品的状态改为已检查，若没有，则显示在列表下方并标识。每扫描一个产品后，若未结束，则重复该步骤 4。

步骤 5：工作人员将所有产品扫描完后，点击结束，移动端显示信封检查的结果。

步骤 6：工作人员选择生成收货履历。移动端获取当前收货时间等信息，从缓存中获取之前下载的所有先期电子履历，从而批量生成收货履历文件。

步骤 7：工作人员选择确定上传，移动端将所有收货履历文件发送给子服务器，子服务器再将履历文件发送给中央服务器。若中央服务器存储成功，则返回消息给子服务器。若子服务器存储成功，则返回消息给移动端。移动端更新本地数据库。

4.3.1.3 一般履历生成流程

除了出生履历和收货履历以外的一般履历的生成过程都是一致的，需要嵌套先期履历文件并且需要单独生成。

具体流程如图 4.10 所示：

步骤 1：工作人员在移动端打开履历生成界面，扫描包含产品序列号的二维码；移动端将序列号发送给子服务器，子服务器搜索其本地事件盒，返回对应的事件数据或未找到；

步骤 2：移动端接收信息，若有事件数据，则将显示具有事件数据的事件信息编辑界面；若无事件数据，则由工作人员选择事件类型，显示对应的空白事件信息编辑界面。工作人员可以通过编辑界面，手动输入出生事件信息进行完善，最后选择确认生成，若取消则跳回步骤 1 重新扫描。

步骤 3：移动端将产品序列号发送给子服务器，子服务器搜索其本地数据库，查找该产品的先期履历文件，若未查找到，则子服务器向中央服务器发起请求，中央服务器在其本地数据库搜索成功后返回履历文件，子服务器再将先期履历返回给移动端。

步骤 4：移动端根据事件信息和先期履历生成对应履历文件，并通过数字签名对履历文件进行签署以保证其完整性，并显示新履历预览界面。

步骤 5：若工作人员选择取消上传，则跳回步骤 2 重新对事件信息进行编辑；若确定上传，则移动端将新的履历文件发送给子服务器，子服务器再将履历文件发送给中央服务器。若中央服务器存储成功，则返回消息给子服务器。若子服务器存储成功后再返回消息给移动端。移动端接收到成功的消息后更新本地数据库。

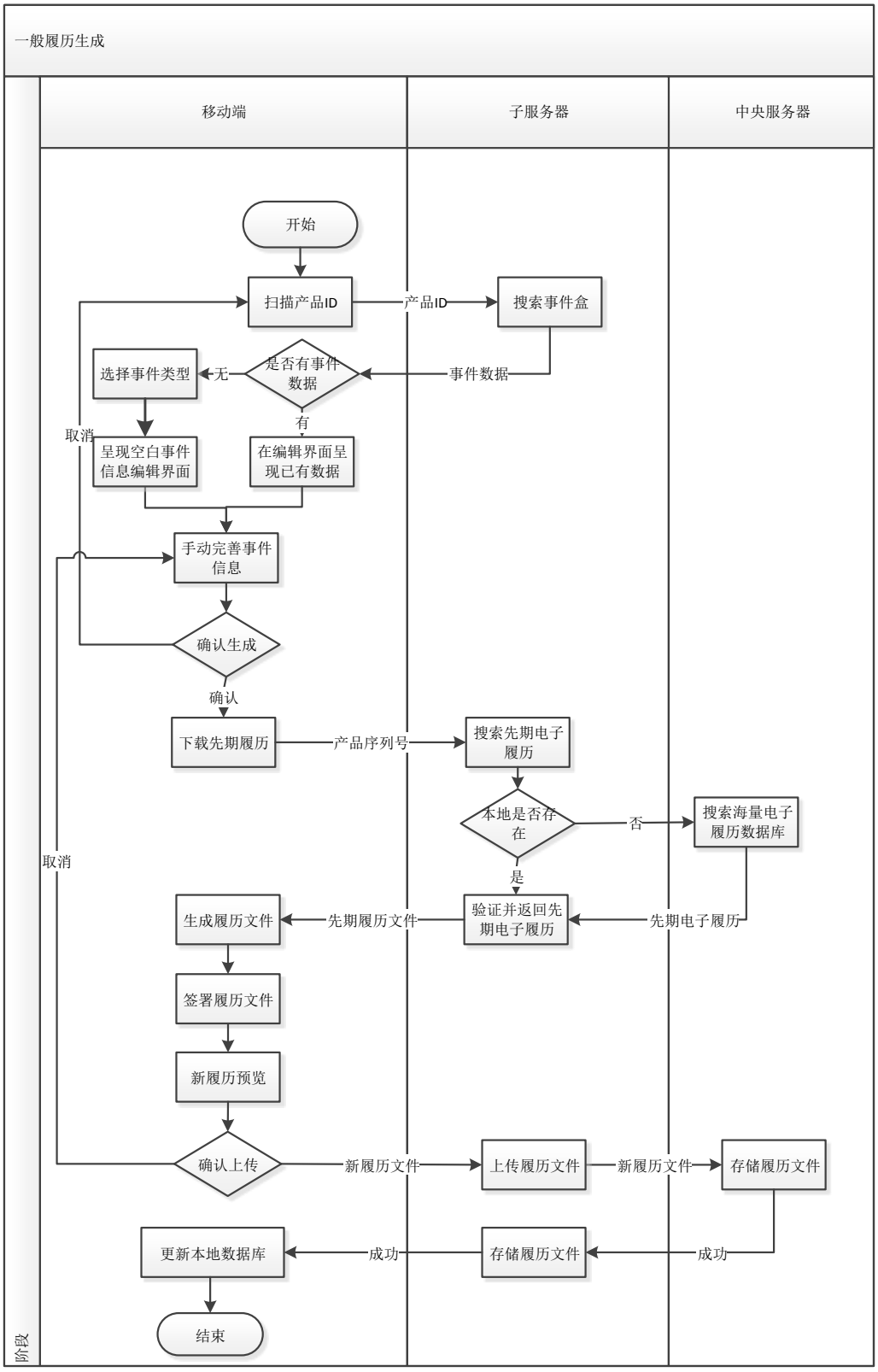

图 4.10 一般履历生成流程图

4.3.2 信封管理流程

信封管理模块是本系统重要的业务功能模块。信封包含了产品集合的序列号信息，实现了传输过程中多个产品的打包集中式管理。具体流程是：在某次传输

过程中，发送方依次扫描产品集合（通常为发往同一接收方的同一批次产品）内的多个产品，生成包含所有产品序列号的信封文件，并将载有信封文件 ID 的二维码标签随该批次产品一起发给接收方；接收方在收到这一批次产品时，通过扫描信封二维码，查看信封内容进行确认，并逐个扫描该批次产品，检查信封内产品是否存在问题，是否有漏发、错发等情况。

以下两个信封管理子模块分别描述了信封生成和信封检查的具体流程和操作。

4.3.2.1 信封生成流程

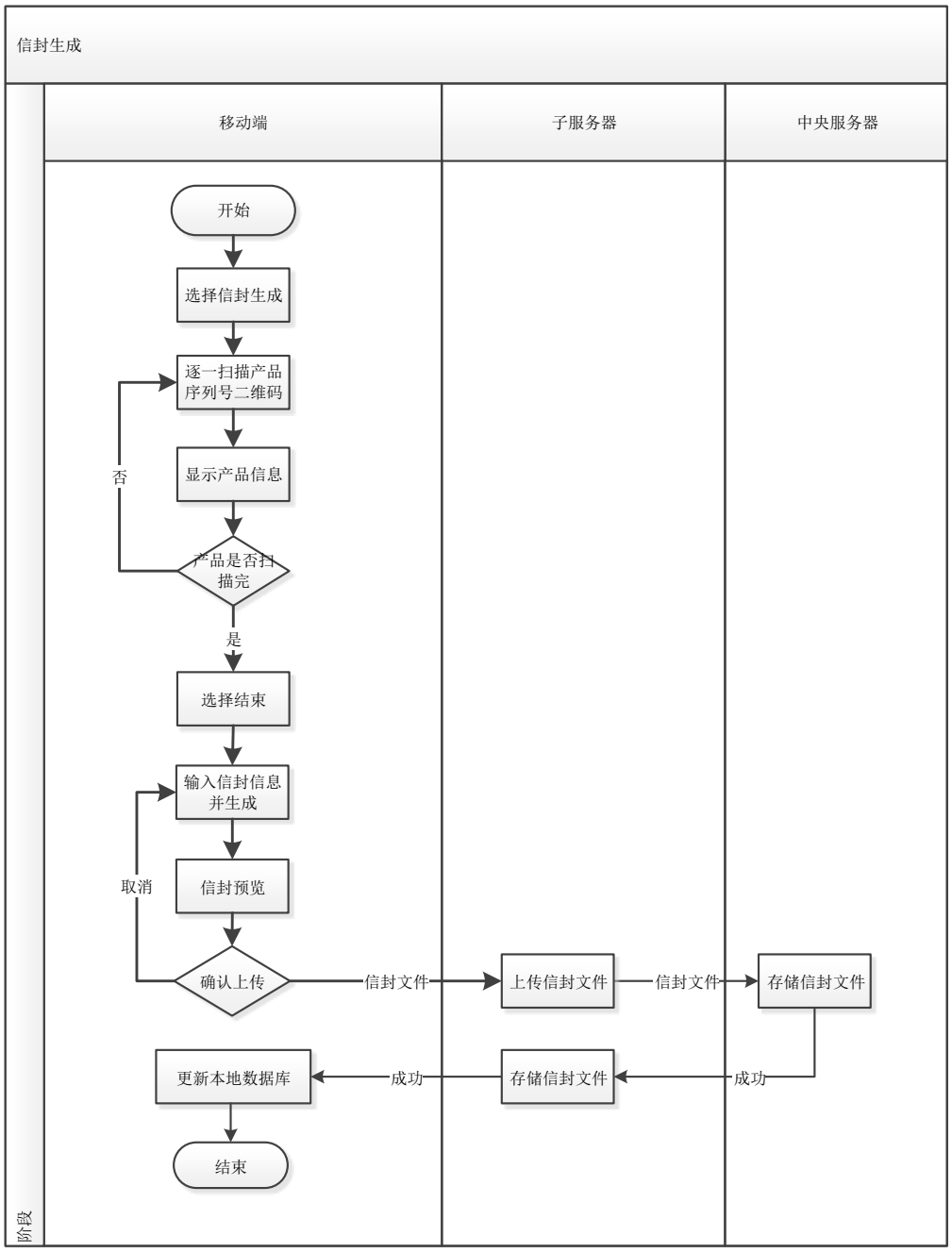

图 4.11 信封生成流程图

信封的生成一般发生在发送方准备将某一批次产品发往下一个接收方时。对于待发货的每一批次的产品集合，均可以对其生成对应的包含集合内所有产品的序列号的信封。产品集合一般为发往同一接收方的同一批次产品。

具体流程如图 4.11 所示：

步骤 1：工作人员在移动端打开信封管理界面，选择信封生成；

步骤 2：工作人员逐一扫描产品集合内的待发货产品，移动端向信封中添加当前产品的序列号并显示当前产品信息。若集合内的产品未扫描完，则继续重复该步骤 2。

步骤 3：工作人员将所有产品扫描完后，点击结束，移动端显示信封生成信息编辑界面。

步骤 4：工作人员输入信息后点击生成，移动端根据所有产品序列号和信封生成信息，生成信封文件

步骤 5：工作人员预览无误后选择确定上传，移动端将信封文件发送给子服务器，子服务器再将信封文件发送给中央服务器。若中央服务器存储成功，则返回消息给子服务器。若子服务器存储成功，则返回消息给移动端。移动端更新本地数据库。

4.3.2.2 信封检查流程

信封检查一般发生在接收方收到某批次产品，且该批次产品附带了信封二维码时。接收方可以通过扫描二维码获取信封 ID，从而查看信封内容并对该批次产品集合进行检查，可以方便快速地检查该集合内产品是否存在问题，并可用于对该集合批量地生成收货履历。

具体流程如图 4.12 所示：

步骤 1：工作人员在移动端打开信封管理界面，选择信封检查，扫描包含信封 ID 的二维码；

步骤 2：移动端将信封 ID 发送给子服务器，子服务器搜索其本地数据库，查找信封文件，若未查找到，则子服务器向中央服务器请求，中央服务器搜索成功后返回文件，子服务器再将信封文件返回给移动端；

步骤 3：移动端将信封文件保存至缓存，并将信封内产品序列号列表显示在界面上。

步骤 4：工作人员开始依次扫描收到的产品，如果该产品在列表中存在，则列表中该产品的状态改为已检查，若没有，则显示在列表下方并标识。每扫描一个产品后，若未结束，则重复该步骤 4。

步骤 5：工作人员将所有产品扫描完后，点击结束，移动端显示信封检查的结果。

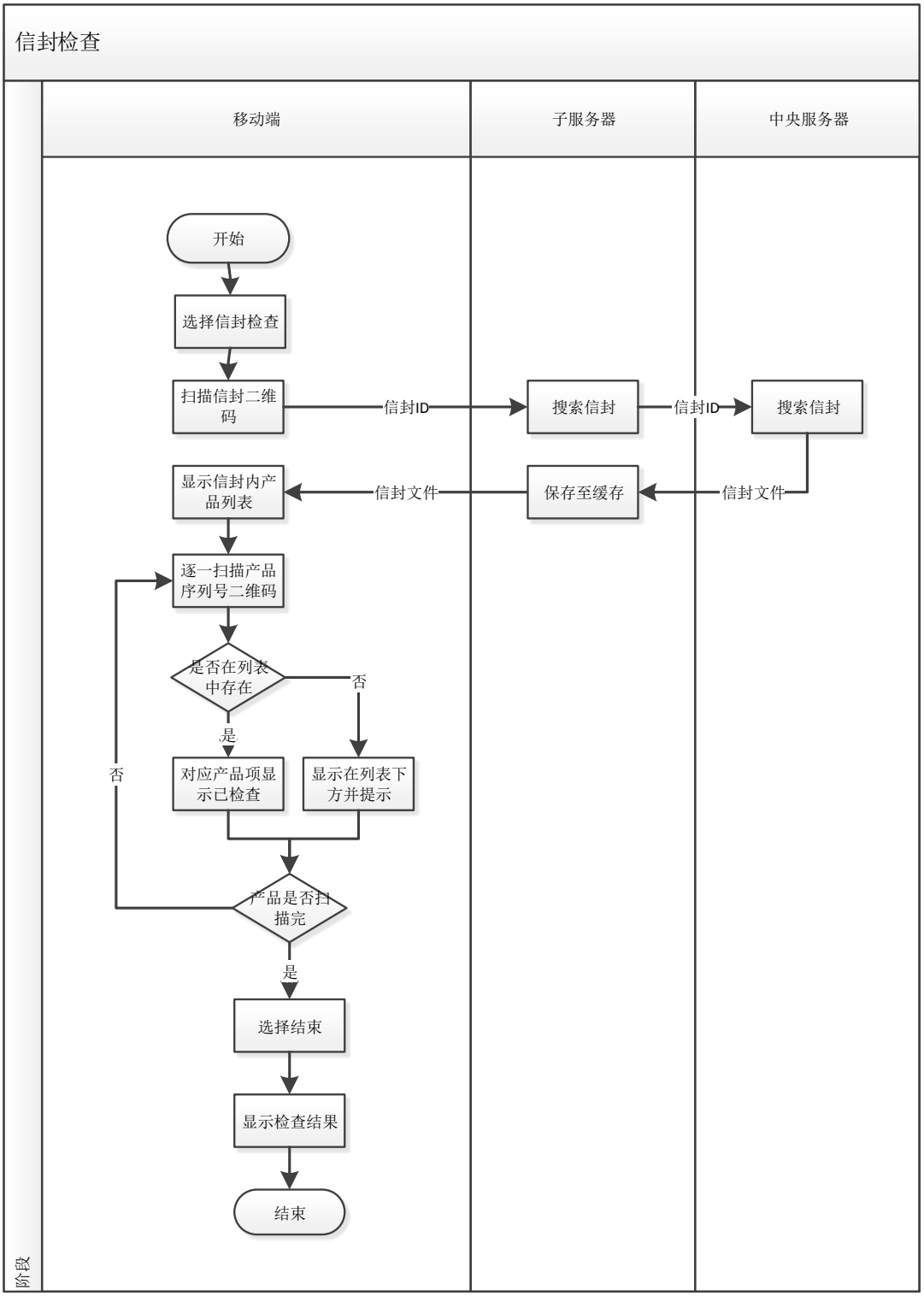

图 4.12 信封检查流程图

4.4 移动电子履历系统的优化

如前文所述，电子履历系统的主要功能包括履历或信封文件的生成、解析、和签名等，而这些功能对移动端的运算能力有一定的要求，同时履历文件的保存也需要移动端具有足够的存储空间，数据的传输也需要耗费一定时间，而且履历

系统还需要重点关注数据的安全保护。然而，由于移动端设备在空间存储能力、计算能力等性能方面具有一定局限性，而且，由于 Android 平台是一个非常自由开放的平台，Android 框架在安全性方面发展得还不够成熟，存在一些安全威胁和漏洞，使得本论文的应用程序在安全性方面面临一些挑战。性能和安全性这两大方面是亟待解决的关键问题。

本论文考虑到移动端的特性，针对履历系统各个功能模块、业务流程的特性，以及相关数据的特征，进行分析，提出一系列对应的改进措施，有的是性能进行提升的，有的是提高安全性的，有的两者兼顾。

4.4.1 履历数据特点分析

4.4.1.1 局部性

对于本履历系统来说，对履历文件的数据访问具有较明显的本地性和局部性特征（Locality），履历文件的联系较为紧密，对其访问的预测有迹可循。一是对于同一信封内的所有履历文件，会在短时间内被访问到；二是生成的履历文件很可能会被继续使用，包括生成后进行查看，或者利用之前生成的履历文件作为先期履历再生成新的履历，特别是对于“收货-加工/打包/其它-发货”这种模式的流程，很可能这几个环节都发生在同一个移动端设备上，故可能连续在同一个设备上生成履历；三是搜索和查看过的履历很有可能会被继续查看，或者用来生成新的履历。

4.4.1.2 操作唯一性

在本履历系统中，因为同一件产品在同一时间只会处于供应链中的一个环节，故同一份履历在同一时间，物理上只可能被一个设备进行修改，易于维护，也减轻了保证数据一致性的难度。

4.4.2 优化措施

4.4.2.1 缓存池

在移动电子履历系统中，一般流程下，如果在本地不存在对应的履历文件，那么履历的搜索和查看需要先向子服务器端请求查询，再将查找到的履历文件下载至本地，另外新履历的生成需要用到先期履历，也需要先从子服务器端进行先期履历文件的下载。而履历文件的下载传输过程会耗费一定的时间，思考是否可以采取缓存池的措施，将每次下载或者生成的履历文件都保存到手机本地，以方便下次需要用到相应履历文件时，能快速地从本地获取，节约传输的时间成本。

更进一步，考虑到移动端设备的存储空间有限，如果每次生成或者下载下来

的履历文件都永久保存在手机，那么随着时间的推移，手机上履历占用的存储空间会越来越多，耗费了过多的空间，而其实本地存储的大多数履历可能不会在短时间内被用到，从而造成了空间的浪费。因此，需要对缓存池的大小进行限制，并且选择一种行之有效、适合本系统的缓存池替换算法策略，来对缓存池中的履历文件进行管理，决定新履历加入缓存池时，替换掉哪些履历（保留哪些履历，删除哪些履历）。

1) 缓存时机

根据前文分析提到的履历数据的本地性特征，主要考虑在以下几个情况下进行缓存：一是当工作人员获取信封后，将信封内的所有的履历下载至本地缓存池中，以便对信封内的履历进行后续操作；二是当生成一个新履历后，将该履历保存至本地缓存池中，以便后续进行查看或继续生成新履历；三是当第一次搜索和查看一个履历后，将从服务器端下载的对应履历文件保存至本地缓存池中，以便之后继续查看或者用来生成新履历。

2) 缓存大小

根据内存大小、履历大小和单批工作的实际情况分析决定，由于一般履历文件大小为几 KB 至几十 KB 而一般手机的内存在 1GB 以上，单批产品可能为几十件到几百件，故在本系统中暂时将缓存大小限制为 100MB。之后工作人员可以根据手机的实际配置情况，进行个性化设置。

3) 缓存替换算法

缓存池的大小是有限的，故需要一个行之有效的缓存替换算法。目前关于缓存替换的研究已经非常成熟，具有多种算法，而本论文需要从这些算法中挑选出简单易行、适合本系统的替换算法，将其应用到本系统中。比较经典常用的缓存替换算法包括：先进先出算法 FIFO（First In First Out），随机置换算法（Random），对象大小算法（Size），最少使用频率算法 LFU（Least Frequently Used），最近最少使用算法 LRU（Least Recently Used）。FIFO 算法是按照时间顺序，最先保存的履历会最先被替换掉，Random 随机选取履历进行替换，这两种算法都过于简单，没有体现出履历数据的局部性特征。而 Size 算法是替换掉最大的文档，以换取较大可用空间，但单个履历文件的数据不是很大，差距也不算大，且这一策略可能导致缓存中一直存在一些较小却再也不会被访问的履历，造成“缓存污染”。LFU 算法是替换掉访问次数最少的履历，但可能导致有些之前具有很高的访问次数而后来不再使用到的履历一直无法被移除，造成“缓存污染”，而且履历访问的次数相差不大，可能造成访问次数最少的履历同时存在多个，无法进行合适的选择。LRU 是替换掉最近被访问最少的履历，考虑到履历数据的局部化特征，同一履历很可能在较短时间内被多次访问，故本系统采取 LRU 算法进行缓存替换。

4) 过期策略和更新策略

因为履历的查询都是根据产品序列号进行的，保存在缓存中的履历文件有可能与中心服务器上的履历文件版本不一致，比如：同一件产品在供应链上经历某个设备结点 A 后转回到 A，但此时在 A 的缓存中的履历还未被替换掉，仍为旧版本的履历。为了保证数据的一致性，每次使用前向中心服务器确认履历的版本号，如果一致则说明该履历未过期，可以直接从缓存池中获取，如果服务器上的版本号比本地的新，则下载并更新本地的缓存。同时，设置隔一段时间（一周）后履历自动失效。此外，当生成新的履历文件后，需要对缓存池中对应的履历进行更新。

4.4.2.2 消息队列

本系统设计的目的之一是为了能够应用于真实的产品供应链场景中，而在实际情况下，在各个流水线环节中，产品都是以批次到达的，故工作人员需要使用本系统在短时间内对多个产品进行处理，比如批量下载或者生成履历文件。而从服务器下载履历文件或者在本地生成履历文件都需要耗费一定运算能力和传输时间，为了使同时批量进行时的过程更加流畅，使工作人员不需要等待网络传输或者机器运算，防止发生阻塞，本系统采用新开线程的方式使其在后台运行，不影响工作人员在前台的操作，并且考虑到过多的线程反而会降低性能，且管理复杂，本系统采取消息队列的机制，保证同时只有主线程（UI 进程）和一个任务线程。

1) 使用场景

场景一：一批产品到达公司，收货员甲要收 100 个货物，他可以连续、快速扫描 100 个二维码，在他扫描的同时，应用会在后台依次生成并更新这 100 个履历，新履历生成的计算并不会阻塞该收货员的扫码行为；当 100 个二维码扫描完毕之后，可能后台才更新了 50 份履历，更新进度会一直以进度条的形式显示在屏幕下方，当全部履历更新完毕之后，方可退出该界面，同时提交服务器。

场景二：正如前文中提到，当工作人员获取信封信息后，需要对信封内的所有履历进行下载并保存至缓存，应用会在后台依次向服务器发起请求，接收到下载的数据文件并保存。

2) 实现机制

结合之前的分析来看，本系统中的消息队列的实现机制如下：首先定义一个队列 Q，用来依次存储当前等待执行的任务；当主线程（UI 线程）触发生成履历或者下载履历的操作任务时，会将该需要执行的任务插入到队列 Q 的队尾，启动一个 Worker 任务线程（W 线程）；W 线程会隔一段时间检查一下队列 Q，如果 Q 为空，则继续睡眠，否则取出 Q 中的队首的第一个任务执行，执行完毕之

后通知 UI 线程更新当前进度。如果 Q 中仍存在任务，则继续执行。W 线程的睡眠时间被设置为一个很短的时间，其存在的意义是防止陷入死循环。

4.4.2.3 延后上传

在本系统中，经常存在“收货-加工/打包/其它操作-发货”这种模式，且很可能这几个环节都发生在同一个移动端设备上，使得需要在同一个设备上连续地生成履历。而中心服务器端只需要存储最新的履历，但每次履历生成后的上传都会占用一定的网络传输时间和运算能力，考虑到这种情况，工作人员可以选择延后上传，也就是说在生成第一个履历时不需要立即上传履历，而是在该台设备上的几个履历生成环节都完成后，再上传最新的履历文件。

4.4.2.4 证书加密下载和加密存储

用于履历签署的证书和相关文件由服务器端分发至手机上，在工作人员第一次登录并使用该系统时，移动端会向服务器端发起请求，打包下载所需证书文件，由于所需文件较大，传输和解压会耗费较多时间，故会将其存储在手机本地，以便之后快速地进行履历签名。考虑到安全性需求，证书采取工作人员的登录密码进行加密后传输至本地，本地再进行解密。并且，在之后每次退出应用程序系统对证书文件进行加密，在登录系统时进行解密，防止手机被其他人获取时，从手机内存中获取到明文的证书文件。

4.4.2.5 传输编码、压缩和加密

履历文件是转化为字节流（byte 数组）进行传输，本系统采取了 Base64 对其进行编码和解码，对数据进行转换使其更规范、更适合传输。另外，由于履历文件的上传和下载过程会占用一定时间，履历传输之前会先对编码后的字节流进行本地压缩，再在接收时进行解压，从而缩减传输的数据大小，节约传输时间，并保证履历文件的完整性。进一步出于安全性的考虑，使用 AES 算法进行加密和解密，具体实现为：算法由 Android 中自带的 javax.crypto.Cipher 包提供，其密钥使用当前登陆的工作人员的用户名和密码，以保证数据的机密性。

第五章 移动电子履历系统的实现

5.1 开发环境配置

为了实现移动电子履历系统各功能模块的开发，开发环境配置如表 5.1 所示。

表 5.1 开发环境配置

类别	配置
移动端操作系统	Android 平台
移动端硬件设备	带有照相功能的 Android 手机、平板等移动设备
开发环境	windows7 操作系统
开发工具	Eclipse，ADT，Android SDK
开发语言	Java
移动端数据库	SQLite

5.2 代码实现

5.2.1 通信方式

5.2.1.1 移动端与服务器端的交互

目前 Android 与服务器进行数据交互的主要方式包括 WebService 和 HTTPService。HTTPService 是基于 HTTP 协议通过 POST 和 GET 等请求进行数据传输，在 Android SDK 中，集成了 Apache 的 HttpClient 工具库来实现 HTTP 服务，模拟 HTTP 请求，该方式传输的数据类型简单。WebService 是基于 SOAP 的跨编程语言和跨操作系统平台的远程调用标准。SOAP（简单对象访问协议）是一种简单易用的、轻量级的、基于 XML 的协议。通过 WebService 可以方便地实现异构系统、不同平台间的数据交换，通过 XML 标准进行数据封装，传输简单或复杂的数据类型。

考虑到本系统需要传输电子履历文件，该文件的数据类型复杂同时表现形式为 XML 文件，同时需要很好的跨平台性，故采取 WebService 来实现本系统的移动端与服务器端的通信。

由于在 Android SDK 中没有提供调用 WebService 的库，而面向 PC 平台的 WebService 工具库，比如 Axis2，CXF 等，都过于庞大，难以移植。因此，本系统考虑使用适合移动设备的第三方类库（KSOAP2）来调用 WebService。

本系统实现了一个叫做 MyWebService 的类，专门负责移动端与服务器端之间的数据交互，通过部署 Web 服务来实现，在 Android 环境下通过第三方类库 KSOAP2 调用 WebService，遵循基于 SOAP 协议的远程调用标准。MyWebService

类中提供了几个函数接口，具体实现的功能如表 5.2 所示。

表 5.2 MyWebService 类中函数接口描述

函数名	参数	返回类型	功能描述
getCompanyInfo()	String url	boolean	移动端向 url 地址发送连接请求，若成功接收到对应子服务器端返回的公司信息，则保存信息并返回 true，否则返回 false。
getLoginInfo()	String username, String password	boolean	移动端向子服务器端发送用户名和密码，请求进行身份验证，若子服务器端返回的认证信息通过，则对返回的个人信息进行存储并返回 true，若不通过则返回 false。
getPedigree()	String product_id	String[]	移动端向子服务器端发送产品序列号，请求下载电子履历，返回子服务器端查询到的电子履历 ID 和文件字节流。
getEnvelope()	String envelope_id	String[]	移动端向子服务器端发送信封 ID，请求下载信封，返回子服务器端查询到的信封文件字节流。
uploadPedigree()	String file_data, String product_id	boolean	移动端上传电子履历文件字节流和产品序列号，返回子服务器端是否保存成功。
uploadEnvelope ()	String file_data	boolean	移动端上传信封文件字节流，返回子服务器端是否保存成功。
getPedigreeID()	String product_id	String	移动端向子服务器端发送产品序列号，返回该产品当前最新的履历 ID。

MyWebService 类中的各个函数都是通过调用 KSOAP2 提供的接口来实现数据上传或者下载的，其关键代码如下：

```
// 创建 SoapObject 对象，指定 WebService 命名空间和调用的方法及其参数；
SoapObject rpc = new SoapObject(nameSpace, methodName);
rpc.addProperty(param, value);
// 设置调用 WebService 方法的 SOAP 请求信息；
SoapSerializationEnvelope envelope =
  new SoapSerializationEnvelope(SoapEnvelope.VER10);
envelope.bodyOut = rpc;
envelope.setOutputSoapObject(rpc);
// 创建 HttpTransportSE 对象，调用 WebService，获取返回的数据；
HttpTransportSE transport = new HttpTransportSE(endPoint);
transport.call(soapAction, envelope);
SoapObject object = (SoapObject) envelope.bodyIn;
```

5.2.1.2 Activity 间的跳转与交互

Android 中，Activity 是用来提供用户与应用程序之间进行交互的接口的应用组件，可以显示界面视图，放置各种控件。由于本系统中，用户和应用程序的交互十分频繁，故本系统实现了多个 Activity 来提供给用户可交互的可视化界面，并且多个 Activity 之间需要进行跳转，同时传递一些数据。

本系统实现的 Activity 具体描述如表 5.3 所示。

表 5.3 系统各个 Activity 的描述

Activity 名称	功能
MainActivity	主界面，提供履历生成、履历浏览、信封管理、个人信息、设置各功能的菜单入口。
SettingActivity	设置界面，提供给用户进行连接子服务器端的设置，同时显示设置信息和公司信息。
LoginActivity	登录界面，提供给用户输入用户名和密码进行登录。
InfoActivity	个人信息界面，显示当前用户的个人资料信息。
SearchActivity	履历查找界面，提供文本输入或二维码扫描的选项。
TextSearchActivity	文本输入界面，提供给用户键盘输入产品序列号进行查询。
QRScanActivity	二维码扫描界面，用来通过摄像头获取图像并进行二维码识读，显示二维码包含的信息。
ShowActivity	履历展示界面，以倒叙列表的形式显示履历信息。
GenerateActivity	履历生成界面，提供单个生成或打包生产的选项。
EnvelopeActivity	信封管理界面，提供信封浏览、信封检查、信封生成的选项。
EnvelopeShowActivity	信封展示界面。以列表形式显示信封内所有产品的信息。
EnvelopeCheckActivity	信封检查界面。以列表形式显示信封内所有产品的检查结果。
EnvelopeGenerateActivity	信封生成界面。显示信封信息编辑界面，提供给用户进行信封生成所需信息的输入和编辑，并提供信封生成接口。
EventInfoActivity	事件信息编辑界面。根据不同的事件类型（出生、初始、、检验检疫、重包装、加工、运输、发货、收货等），显示对应的控件和界面，提供给用户进行对应履历生成所需信息的输入和编辑，并提供履历生成接口。

为了实现各个功能流程，各个 Activity 之间需要进行跳转，同时还需要传递一些数据，以便下一个 Activity 做出相应的操作。比如：对于履历生成的功能来说，首先需要扫描产品上的二维码标签，然后对事件信息进行编辑，最后对履历文件进行查看，即需要的跳转顺序以及传递的参数为：主界面——履历生成界面——（传递履历生成操作类型，"type": "epedigree_generate"）——二维码扫描界

面——（传递产品序列号，”product_id”: product_id）——事件信息编辑界面——（传递产品序列号，”product_id”: product_id）——履历展示界面。

Activity 之间通过 Intent 对象来实现跳转和交互，通过 Intent 的 Extras 属性以键值对的形式存储需要传递的信息，比如由 CurrentActivity 跳转至 OtherActivity，在 CurrentActivity 中的关键代码如下：

```
// 创建一个 intent 对象；
Intent intent = new Intent();
intent.setClass(CurrentActivity.this,OtherActivity.class);
// 以键值对的形式将所需传递的数据存储在 Extras 中；
intent.putExtra(key, value);
// 启动新的 Activity，实现跳转；
startActivity(intent);
```

而在新启动的 OtherActivity 中，如果想获取传递的数据，则实现代码如下：

```
// 获取当前的 intent，取得其 Extras，根据 key 值获取对应的 value；
Intent intent = this.getIntent();
String value = intent.getExtras().getString(key);
```

5.2.2 数据存储

本系统中，需要存储的数据包括用户个人信息，公司信息，密钥、履历和信封的基本信息，密钥、履历和信封文件。针对不同的数据类型，本论文需要对其选择合适的数据存储方式。

在 Android 中，共有四种比较常见的存储方式：SharePreference、File、SQLite 以及 Content Provider。SharedPreference 本质是一个 XML 文件，主要用于以键值对的形式，存储比较简单的数据。File 是将数据以文件的形式保存在存储空间中。SQLite 是一个轻量级的数据库，支持基本 SQL 语法，方便查询。ContentProvider 主要用于两个系统之间的数据共享。

对于较为简单的用户基本信息和系统配置信息，本系统采取 SharedPreference 存储方式将其以键值对的形式保存在一个 XML 文件中，并将该文件存储在“data/data/程序包名”私有目录下，使得该数据只对本应用程序可见，不会被其它应用程序访问到。

对于密钥、履历和信封文件，本系统采取 File 存储方式。由于密钥文件较小且需要安全保护，将其存储在容量有限的内部存储空间，即“data/data/程序包名”私有目录下，受到访问权限控制，更加稳定、安全；另外，由于履历和信封文件所占空间较大且需要进行方便的访问，故将其存储在容量充足的外部存储空间，即 SD 卡上。

对于密钥、履历和信封的基本信息，为了进行方便的查询、添加、更新和管理，本系统采用 SQLite 数据库来进行存储，可以很方便地通过 SQL 语句对数据进行操作。

5.2.3 二维码

本系统采用 Google 提供的开源 ZXing 项目来实现二维码的识读功能，通过引入 com.google.zxing 源代码和 jar 包进行开发，在 QRScanActivity 类中实现了二维码的扫描和识读。首先在 AndroidManifest.xml 中添加了关于摄像头的访问权限，具体如下：

<uses-permission android:name="android.permission.CAMERA" />

<uses-feature android:name="android.hardware.camera" /><!-- 使用照相机权限 -->

<uses-feature android:name="android.hardware.camera.autofocus" /><!-- 自动聚焦权限 -->

权限配置成功后，通过 onResume()函数中调用 initCamera()初始化摄像头，同时初始化扫描界面。然后，通过摄像头实时地获取当前拍摄到的画面的图像，并且在后台实时地对图像进行二维码识别和解码。当扫描成功后，即识别出当前图像包含二维码且对二维码解码成功后，能够得到二维码中包含的全部信息，同时会调用 handleDecode()函数，可以在该函数体内进行后续的操作，比如对识别出的信息进行一些处理，或者跳转到下一个界面。

5.2.4 加密算法

本系统采用 Android 自带的 javax.crypto.Cipher 工具包，利用 Cipher 提供的多种加密算法来实现对数据的加密和解密。

实现加密的具体代码如下：

```
// 实例化，设置加密算法，比如 AES；
Cipher cipher=Cipher.getInstance(CIPHER_ALGORITHM);
// 初始化，设置为加密模式；
cipher.init(Cipher.ENCRYPT_MODE, key);
// 对数据执行加密操作；
byte[] decode_data = cipher.doFinal(data);
```

对应地，实现解密的具体代码如下：

```
Cipher cipher=Cipher.getInstance(CIPHER_ALGORITHM);
cipher.init(Cipher. DECRYPT_MODE, key);
byte[] data = cipher.doFinal(decode _data);
```

5.3 界面实现

本系统面向的用户分角色包括工作人员和消费者。本系统具体实现为一个 Android 应用程序。对于用户来说，需要手持一台 Android 操作系统的移动设备（比如 Android 手机、平板等），在设备上安装并打开该应用程序，主要的互动交流方式是通过对其手持的移动设备进行操作来实现。和用户相关的界面实现，大部分都体现在 Android 应用界面上。

具体表现为：

用户通过移动设备的摄像头扫描二维码标签；

用户通过键盘输入文本内容，比如产品序列号、事件信息、用户名及密码等；

用户通过点击 Android 应用界面上的按钮等控件选择相应功能；

系统将展示的结果比如履历信息、信封内容等输出在设备屏幕界面上。

以下界面实现都基于用户手持一台联网的移动设备，下载并安装本移动电子履历系统的 Android 应用程序，点击图标打开应用。

5.3.1 消费者界面

消费者无需进行登录，可以对根据序列号对履历进行查询。本系统为消费者提供了履历查询的界面，如图 5.1 所示，并提供了两种输入序列号的方式：通过摄像头扫描包含产品序列号信息的二维码标签；通过键盘文本输入产品序列号。用户可以通过从商品的包装或者其他方式获取二维码标签或者产品序列号。

用户点击履历查询，选择二维码扫描或者手动输入产品序列号。若选择二维码扫描，则跳转至扫描界面，系统打开设备自带摄像头拍摄当前画面，扫描到二维码并成功获取序列号后，跳转至显示界面返回查询结果；若选择手动输入，则文本框中输入序列号后，点击搜索图标，跳转至显示界面显示该产品的履历信息。

图 5.1 履历查询界面

5.3.2 工作人员界面

对于工作人员来说，主要操作包括：初次使用该系统时进行初始化设置；用户登录；履历管理；信封管理。

5.3.2.1 初始化设置

应用程序在第一次被使用时，会提供初始化界面，如图 5.2 所示，使得工作人员可以输入服务器的 url 地址以进行连接，连接成功后获取公司信息并保存。并且在之后，也可以在“设置”界面接口中查看公司信息或者对 url 进行修改。

图 5.2 初始化设置界面

5.3.2.2 工作人员登录

初始化设置成功后，应用程序会跳转至登录界面，使得工作人员可以输入用户名和密码进行身份验证，验证成功后跳转至主界面，并可在“我的资料”界面接口查看用户个人信息，并进行注销操作。相关界面如图 5.3 所示。

图 5.3 用户登录和个人信息界面

5.3.2.3 履历管理

工作人员登录成功后可以对履历进行管理：履历查询、履历生成。履历查询和消费者接口中的一样。对于履历生成来说，工作人员首先通过扫描二维码获取产品序列号，然后跳转至事件编辑界面；接着选择履历事件类型，系统根据选择的事件类型的不同，展示对应的事件信息编辑界面，该界面提供了相应的输入和编辑控件，以便工作人员能够通过移动设备，输入或编辑生成对应事件类型的电子履历所需的信息。在信息编辑完成后，工作人员可以点击右上角的“生成”按钮，系统将在后台生成相应履历，并提示是否生成成功。相关界面如图 5.4 所示。

图 5.4 履历生成界面

5.3.2.4 信封管理

如图所示，移动电子履历系统提供了一系列接口，使得工作人员登录成功后可以对信封进行管理，包括：信封读取、信封检查、信封生成。工作人员通过点击“信封读取”进入二维码扫描界面，扫描包含信封 ID 的二维码标签后，应用程序跳转至信封显示界面，展示该信封内所包含的产品信息。界面如图 5.5 所示。

图 5.5 信封管理界面

5.4 系统部署

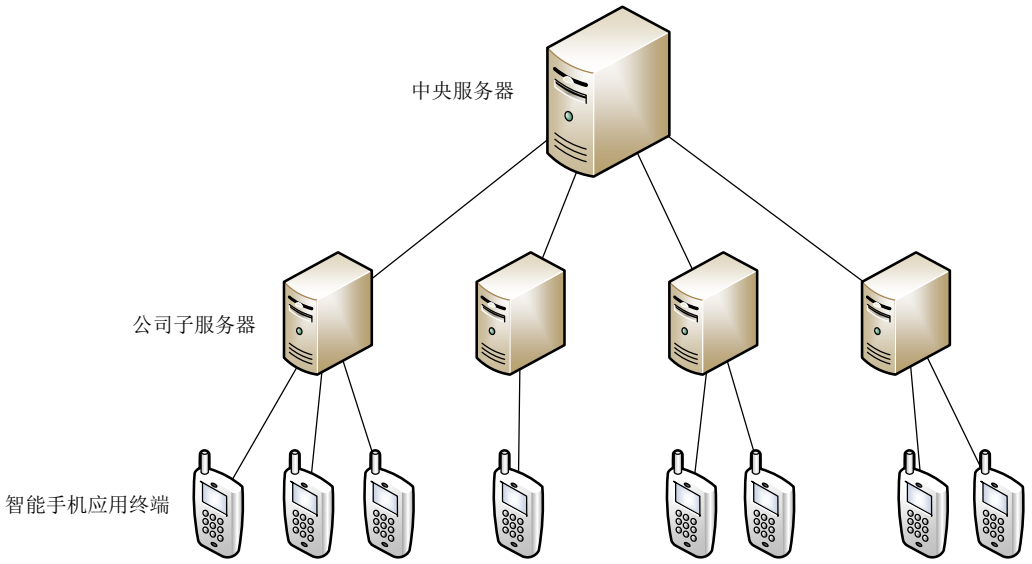

图 5.6 电子履历系统部署图

图 5.6 为移动电子履历系统的部署图。其中，中央服务器单独部署，负责为供应商处的电子履历提供注册和管理功能，同时为用户提供网页端履历查询功能。电子履历子系统则部署在不同的供应商处，负责所属供应商履历的存储、密钥的管理和工作人员的管理。移动端系统部署在工作人员手持的移动设备上，以 Android 应用程序的形式进行部署，负责履历/信封文件的生成、查询和上传等功能。每个子系统可以与多个移动端系统进行通信，移动端不能与中央服务器直接通信，只能通过子服务器间接实现数据交互。系统之间的通信通过 WebService 实现。

第六章 总结和展望

6.1 总结

本论文设计并实现了一个基于 Android 平台的移动电子履历系统，实现了产品在整个供应链流程中的信息化管理，从而便于快速追溯，及时定位问题，提高消费者对产品的信任度。

本论文首先调研了当前的电子履历系统的研究背景和国内外研究现状，分析了移动电子履历系统的研究意义，确立了具体的研究内容。随后介绍了物联网、电子履历和 Android 平台等技术，为移动电子履历系统的研发提供基础。然后，对移动电子履历系统的功能需求和业务流程进行了详细的分析。接着设计了系统的总体框架和各个模块框架，定义了履历和信封的数据模型，设计了主要功能模块的流程，并提出一些针对移动端特性的改进措施。最后，搭建了 Android 应用程序开发环境，开发实现了移动电子履历系统的各个功能模块，并且进行了系统部署。

本系统通过基于Android平台的移动设备，采取二维码识别或其它感知方式，收集、存储并处理产品在各个环节上的信息，遵循设计制定的规范，生成相应的电子履历文件，并且利用数字签名技术进行履历签署，从而保障履历文件的完整性和可信性。同时，该系统向移动端提供了验证和展示产品电子履历的接口，方便消费者快速地查询履历并全面地获取产品相关信息。移动电子履历较传统电子履历系统而言，更加方便快捷易用，并且能够很好地应用到实际中，应对当前高速发展的移动互联网的技术特性，以满足不断扩大的安全追溯需求。

论文的主要工作如下：

1) 本论文调研了当前的电子履历系统的研究背景和国内外研究现状，定义、明确了产品在整个供应链中的业务流程，设计制定了规范的履历标准，设计并实现了基于 Android 平台的移动电子履历系统。该系统实现了产品在整个供应链流程中的信息化管理，从而便于快速追溯，及时定位问题，提高消费者对产品的信任度。
2) 本论文将物联网与移动互联网相结合，将电子履历系统与移动端和移动应用相结合，使人与物、物与物之间的联系更紧密。该系统利用移动终端，基于 Android 平台，针对产品供应链的业务流程，考虑并分析了电子履历系统与移动端和移动应用的特性，提供多维感知方式以及数据的安全保护，充分体现了移动性、便携性和易用性，使用方便快捷，操作简单高效，界面简洁友好，具有很好的实际应用价值。
3) 考虑到移动端设备在存储能力、计算能力等性能方面的局限性，以及

Android 平台和 Android 应用程序在安全性方面发展的不够成熟，存在一些安全威胁和漏洞，本论文根据本系统的具体功能特性和数据特征，提出了一系列改进措施进行性能优化和安全性的提高，以满足电子履历系统在计算能力、存储空间和安全性方面的需求。

6.2 展望

本论文设计并实现的移动电子履历系统具有实际应用价值，但仍然存在挑战，还有一些可以改进的地方。

目前本论文的密钥管理系统相对比较简单，在未来会考虑移动电子履历系统的特性，设计一个更加严密更加安全的密钥管理系统。可以尝试的一个方法是建立一个可信任的第三方机构，专门负责密钥的管理、分发等。

另外，在本论文提到的系统中，选择二维码作为数据载体和感知来源，而二维码的生成更新不太方便，且二维码直接暴露在外，很容易被其它设备读取获取其中的信息。而在之后的工作中，我们将考虑采取 NFC 技术，可以很方便地进行数据的读取和更新写入，并且更具有保密性。

此外，对于移动端带来的性能问题和安全问题，还有很大的提升空间。在未来的工作中，我们会寻找更多的方法来进行改进。

参考文献

[1] Wikipedia．阜阳劣质奶粉事件[EB/OL]．http://zh.wikipedia.org/wiki/阜阳劣质奶粉事件，2014-3-10．

[2] Wikipedia．2008 年中国奶制品污染事件[EB/OL]．http://zh.wikipedia.org/wiki/2008 年中国奶制品污染事件，2014-8-18．

[3] Wikipedia．2013 年台湾食品安全问题事件[EB/OL]．http://zh.wikipedia.org/wiki/2013 年台湾食品安全问题事件，2014-9-11．

[4] 百度百科．毒胶囊事件[EB/OL]．http://baike.baidu.com/view/8384881.htm?fr=aladdin，2014-9-28．

[5] J. Shi，Y. Li，W. He，D. Sim．SecTTS: A secure track & trace system for RFID-enabled supply chains[J]．Journal of Computers in Industry，2012，pp：574–585．

[6] IDC．Android and iOS Continue to Dominate the Worldwide Smartphone Market[EB/OL]．http://www.idc.com/getdoc.jsp?containerId=prUS24676414，2014-2-12．

[7] Number of Android applications．Technical report[R]．AppBrain, 2014．

[8] L. Atzori，A. Iera，G. Morabito．The Internet of Things: a survey[J]．Computer Networks，2010，54：2787–2805．

[9] M. Domingo．An overview of the Internet of Things for people with disabilities[J]．Journal of Network and Computer Applications，2012，35，584–596．

[10] L. Zheng，H. Zhang，W. Han，X. Zhou，J. He，Z. Zhang，Y. Gu，J. Wang．Technologies, applications, and governance in the Internet of Things[B]．In: Internet of Things - Global Technological and Societal Trends. From Smart Environments and Spaces to Green ICT，River Publishers，2011．

[11] EPCglobal．Pedigree Ratified Standard[EB/OL]．http://www.gs1.org/gsmp/kc/epcglobal/pedigree/pedigree_1_0-standard-20070105.pdf，2007-1-5．

[12] 新东阳．台湾新东阳产品履历[EB/OL]．http://www.hty.com.tw/meatclub_resume.php，2014-9-20．

[13] 凯馨宝业．台湾凯馨宝业履历生产[EB/OL]．http://tw.gugugoo.com/resume.html，2014-9-20．

[14]金兰．台湾金兰产品履历[EB/OL]．http://60.251.239.164/klrme/klrme.htm，2014-9-20．

[15]台湾行政院农委会．台湾农产品安全追溯资讯网[EB/OL]．http://taft.coa.gov.tw/，2014-9-20．

[16]味全．味全奶粉履历网站[EB/OL]．http://lvli.weichuan.com.cn，2014-9-20．

[17]K. Ashton．That ‘internet of things’ thing[J]．RFID Journal，2009，22: 97-114．

[18]J. Gubbi，R. Buyya，S. Marusic，et al．Internet of Things (IoT): A vision, architectural elements, and future directions[J]．Future Generation Computer Systems，2013，29(7): 1645-1660．

[19]W. Baoyun．Review on internet of things[J]．Journal of Electronic Measurement and Instrument，2009，23(12): 1-7．

[20]IBM．IBM InfoSphere Traceability Server Documentation[EB/OL]．http://publib.boulder.ibm.com/infocenter/itshelp/v3r0/index.jsp，2012-3-1．

[21]Oracle．Oracle Pedigree and Serialization Manager[EB/OL]．http://www.oracle.com/us/industries/life-sciences/oracle-pedigree-serialization-150362.html，2014-9-20．

[22]Google．Android Developer[EB/OL]．http://developer.android.com/，2014-9-20．

[23]Tim Bray．What Android Is[EB/OL]．http://www.tbray.org/ongoing/When/201x/2010/11/14/What-Android-Is#3112，2010-1-14．

[24]刘云浩．物联网导论[M]．北京：科学出版社，2010．

[25]R. Want．An introduction to RFID technology[J]．Journal of Pervasive Computing，2006，pp：25–33．

[26]H. Kopetz．Internet of things[M]．Real-Time Systems．Springer US，2011: 307-323．

[27]D. Eastlake，J. Reagle，D. Solo．XML-Signature Syntax and Processing (W3C/IETE recommendation)[S]．W3C，2002．

[28]RA. Haraty，ANE. Kassar，B. Shibaro．A Comparative Study of RSA Based Digital Signature Algorithms[J]．Journal of Mathematics and Statistics，2006，vol. 2，no. 1，pp：354-359．

[29]W. Ford，R. Housley，W. Polk．Internet X.509 Public Key Infrastructure Certificate and CRL Profile[S]．RFC 2459，1999．

[30]DM. Lamber，MC. Cooper．Issues in supply chain management[J]．Journal of Industrial Marketing Management，2000，pp：65–83．

[31]W. Han，Y. Gu，W. Wang，et al．The design of an electronic pedigree system for food safety[J]．Information Systems Frontiers，2012：1-13．

[32]EO. Blass，K. Elkhiyaoui， R. Molva．Tracker: Security and Privacy for RFID-based Supply Chains[C]．NDSS，2011．

[33]K. Michael，L. McCathie．The pros and cons of RFID in supply chain management[C]．Mobile Business，2005．ICMB 2005．International Conference on. IEEE，2005: 623-629．

[34]RH. Weber，R. Weber．Internet of Things[M]．Springer，2010．

[35]E. Welbourne，L. Battle，G. Cole，et al．Building the internet of things using RFID: the RFID ecosystem experience[J]．Internet Computing，IEEE，2009，13(3): 48-55．

[36]QB. Sun，J. Liu，S. Li，et al．Internet of Things: Summarize on Concepts, Architecture and Key Technology Problem[J]．Journal of Beijing University of Posts and Telecommunications，2010，3(3): 1-9．

[37]L. Tan，N. Wang．Future internet: The internet of things[C]．Advanced Computer Theory and Engineering (ICACTE)，2010 3rd International Conference on. IEEE，2010，5: V5-376-V5-380．

[38]D. Guinard，I. Ion，S. Mayer．In search of an internet of things service architecture: REST or WS-*? A developers' perspective[M]．Mobile and Ubiquitous Systems: Computing, Networking, and Services．Springer Berlin Heidelberg，2012: 326-337．

[39]P. Giner，C. Cetina，J. Fons，et al．Developing mobile business processes for the internet of things[J]．Pervasive Computing，IEEE，2010，9(2): 18-26．

[40]M. Darianian，MP. Michael．Smart home mobile RFID-based Internet-of-Things systems and services[C]．Advanced Computer Theory and Engineering，2008．ICACTE'08．International Conference on IEEE，2008: 116-120．

[41]MP. Michael，M. Darianian．Architectural solutions for mobile RFID services for the internet of things[C]．Services-Part I，2008．IEEE Congress on. IEEE，2008: 71-74．

致 谢

在本论文完成之际，首先要感谢论文指导老师韩伟力老师，给予我学业方面的支持和帮助，感谢他从论文题目的选定，到系统的设计和实现，到系统的优化和改进，再到论文的写作和修改，都进行了悉心的指导，通过多次讨论和修正，使我顺利地完成了这篇论文。同时感谢王蔚、张胤同学和顾赟学姐在本项目中的贡献以及对我的帮助和指点。

还要感谢复旦大学全体老师的教导和培养，特别是软件学院的老师们，在学习生涯中，给予我的指导和关怀，使我在学业上受益无穷。

衷心的感谢我的父母和朋友们对我的关心、支持和理解，是他们的鼓励和关怀，促使我不断进步。

最后，衷心感谢各位专家百忙之中对本论文的审阅和赐教。

论文独创性声明

本论文是我个人在导师指导下进行的研究工作及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其它机构已经发表或撰写过的研究成果。其他同志对本研究的启发和所做的贡献均已在论文中作了明确的声明并表示了谢意。

作者签名：_____ 日期：_____

论文使用授权声明

本人完全了解复旦大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。保密的论文在解密后遵守此规定。

作者签名：_____ 导师签名：_____ 日期：_____