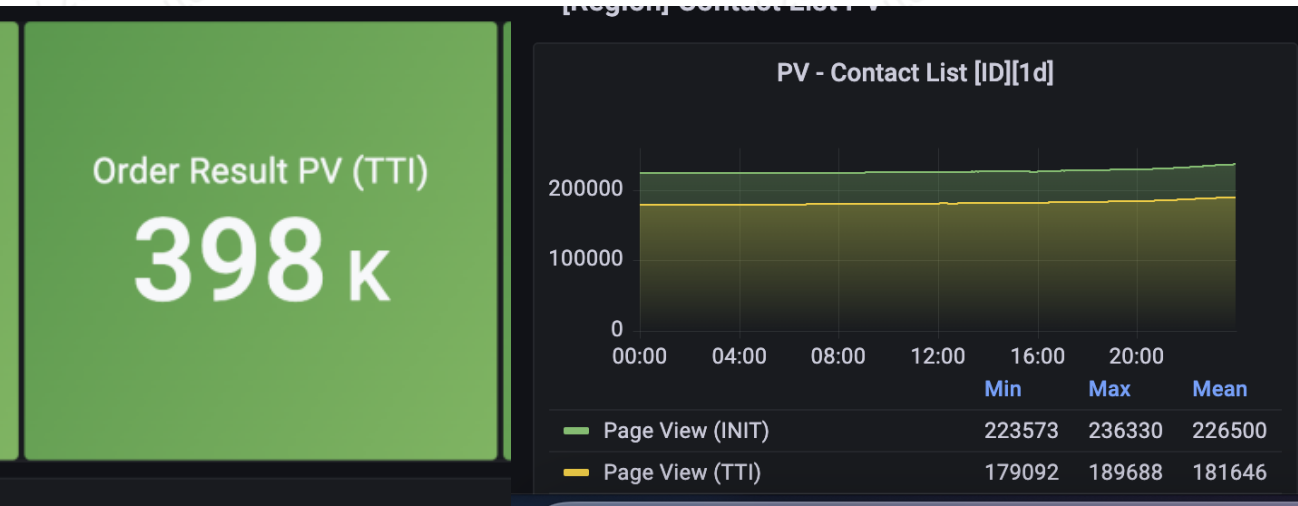


# Improve Contact UX and Performance Report

PRD	Transfer 2.0 - Merge Phase 1: ID QRIS, PH Instapay#43P2PContactListPage
Jira Epic	<a href="#">SPPAY-110326</a> - [Tech][WS][RN] P2P - Improve Transfer To Contact Flow UX & Code Quality <span>REGRESSION</span>
Formatted Phone Number	[TD][FE]Enhance Transfer Contacts Page Performance
FE TD	[Tech][RN] P2P - Improve Transfer To Contact Flow UX & Code Quality
Native TD	[TD] ContactManager SDK New Flow
stage	<span>DONE</span>
Document Owner (Dev)	guankun.chen@seamoney.com qinqin.he@seamoney.com
PM	qianwei.lim@seamoney.com
Document Owner (QA)	

## 1.背景

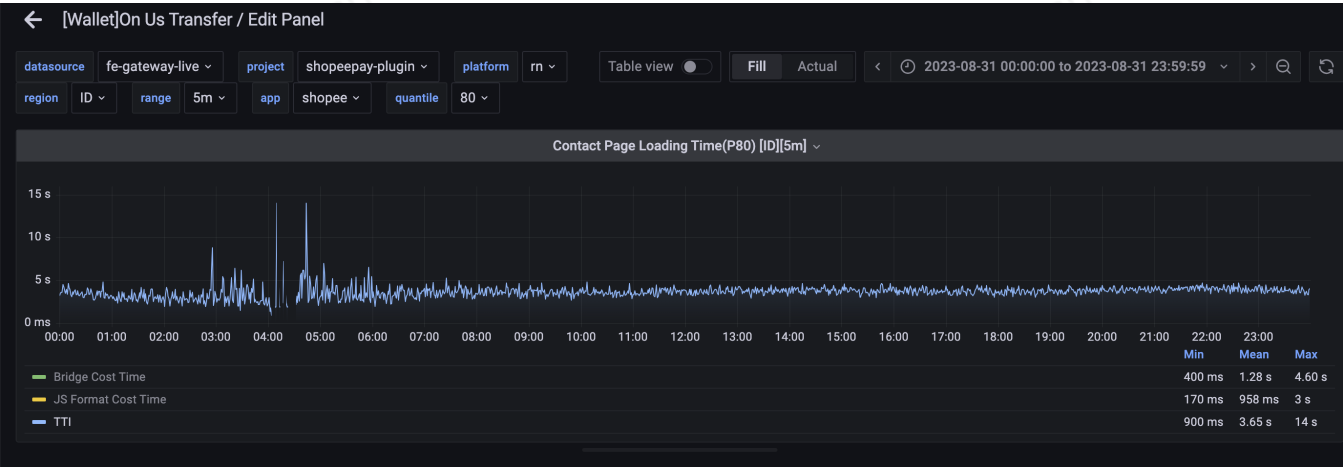
on-us flow是wallet重要的业务之一，每天都有着大量的流量，而contact页面是其中重要的一环，直接影响整个流程的完成率和用户体验。以ID地区2023-8-31 为例，完成on-us flow（以结果页PV为准）有398k，contact页面流量则有236k，通过数据表明用户有很强的意愿通过contact页面对亲朋好友进行转账。



但是目前contact页面存在严重的性能问题，随着用户通讯录增加，页面加载的越缓慢，以下是1000条通讯录打开页面的时长，Local对比竞品的数据报告。可以看到我们的数据十分糟糕。

	ShopeePay	DANA	GoPay	LinkAja	OVO	DOKU
Detect active users	✓	✓	✗	✗	✗	✗
Show profile picture	✓	✗	✗	✗	✗	✗
Loading state	✓	✓	✗	✗	✗	✗
Loading time (iOS, iphone X, 1161 contacts)	5 secs	1 sec	<1 sec	<1 sec	<1 sec	<1 sec

以下是同一天线上contact页面TTI P80数据，用户平均需要3s才能加载完成页面。



通过以上Local报告和线上监控，我们知道了contact页面的重要性，为了提高用户的体验，加快on-us转账流程，我们需要对改页面进行必要的重构。

## 2.目标

SN	Target	Status
1	3000条数据实现秒开	DONE
2	重复数据去重	DONE
3	梳理业务逻辑，降低页面错误率，提高可维护性	DONE
4	区分shopee用户是否在通讯录中	WAITING
5	Wallet Icon正确展示	WAITING

## 3.结果对比

说明：

*Total Time(TTI): 表示用户可以正常看到通讯录列表数据，并可以看到格式化后的手机号;*

*step0: 表示着用户进入页面打开时间节点;*

step1: 表示用户进入页面后，RN开始获取用户通讯录列表数据的时间点；

step2: 表示用户获取到通讯录列表数据，开始对列表进行手机号格式化完成时间点；

Device	Before Optimization				After Optimization					
	Total Time (TTI)	Step0: Page Init	Step1: Contact Bridge	Step2: Phone Format	First Visit: Total Time(TTI)	Non-First Visit: Total Time(TTI)	Step0: Page Init	Step1: First Visit: Contact Bridge	Step1: Non-First Visit: Contact Bridge	Step2: Phone Format
Iphone 11 pro (iOS)	>3s	0.04s	>3s	0.483s	2.6s	0.268s	0.04s	2.206s	0.085s	0.045s
Iphone 8 Plus (iOS)	>3s	0.04s	>3s	0.732s	3.01s	0.734s	0.04s	2.22s	0.288s	0.068s
Galaxy Note 10+ (Android)	>5s	0.04s	>4s	1.78s	3.091s	0.695s	0.04s	2.036s	0.197s	0.13s
RealMe RMX2202 (Android)	>5s	0.04s	>4s	2.032s	3.45s	0.68s	0.04s	2.922s	0.318s	0.15s
RedMe K40 (Android)	>5s	0.04s	>4s	1.96s	3.69s	0.471s	0.04s	2.831s	0.122s	0.175s

ps: contact count is 3000

结论：

- 1. Andorid用户首次进入页面，TTI优化后加快了1s ~ 2s，从>5s → 3s左右提升了67%；IOS用户平均仍在3s左右。
- 2. 用户非首次进入页面，TTI优化十分明显，Android用户>5s → <1s 提升了400%，IOS用户从>3s → <1s 提升了200%。
- 3. 综上，我们实现了contact页面在3000条数据下非首次进入页面的秒开。

4.优化功能列表

SN	Detail	Remark
1	优化用户体验，实现页面3000条数据秒开	<ul style="list-style-type: none"><li>new BridgeSDK 缓存用户通讯录系统数据，该功能减少了TTI 3s左右</li><li>前端从上到下批量格式化用户手机号，每批次为100个手机号，该功能减少了TTI 1.5s左右</li></ul>
2	保证数据准确性	重复手机号，去重，只会展示其中一个
3	梳理业务逻辑，减少bug，增强代码健壮性，降低维护成本	

批量格式化不会阻塞用户交互，用户大量交互批量格式化完成时间为6.5s，页面静止则完成时间为3.26s

5.影响范围

- 用户打开页面后，通讯录数据正常展示，每项包括，头像、联系人名、电话号码和username以及Wallet Icon
- 进入下一步流程为转账，到input amount页面后，receiver展示信息与在contact页面保持一致
- 用户通讯录如果有多个重复手机号，则只会展示其中一个
- 搜索功能，能够区分被搜索的receiver来自 Contacts 或者 Not From Contacts

- 批量格式化手机号功能，如果滚动或搜索通讯录快于批量格式化，则会先展示不带格式化的手机号，等批量格式化完成会自动更新为格式化的手机号
- 传给后端的手机号，以US BE返回的手机号为准；前端格式的手机号只负责列表展示。E.g. 在ID地区，US返回了SG手机号 81902989 未加区号，前端加上区号格式化并展示为 (+65) 8190 2989，在调用后端API ValidateUser、createTicket等等，传给后端的是81902989。

## 6. 变更范围

- Contact BridgeSDK 重构
- contact页面 重构

## 7. 风险评估和预案

SN	Risk	Plan
1	重构可能引进的严重错误 <ul style="list-style-type: none"> <li>▪ contact页面不能展示联系人列表</li> <li>▪ 搜索结果有数据，却不能展示任何搜索结果</li> <li>▪ 未知错误导致页面crash</li> </ul>	SPPAY_WALLET_REFACTOR_CONTACT_LIST  该toggle控制用户进入新老流程
2	多个重复手机号，过滤后只会展示其中一个，可能导致未知出错，e.g. 本来应该展示的联系人被错误过滤掉了。	SPPAY_WALLET_CONTACT_DEDUPLICATION  该toggle控制新流程是否过滤重复手机号  关掉开关则展示所有通讯录数据
3	分页进行格式化手机号逻辑异常： <ul style="list-style-type: none"> <li>▪ 进入页面后，优先格式化200条数据，并展示页面；并静默分页批量处理剩余部分数据；</li> <li>▪ 快速滑动时，格式化数据展示异常 &amp; 是否导致页面滚动异常 &amp; 页面卡死等现象；</li> </ul>	SPPAY_WALLET_CONTACT_PAGINATION  该toggle控制是否批量格式化手机号  关掉开关，则页面初始化展示loading直到格式完所有手机号后才会展示联系人列表