

A | 医生、诊所与上门准入清单

每人一档、每地点一档、每基金一项认可。证据原件由诊所留存，平台记录有效期与核验人。

核验对象	证据与判定	入口控制
AHPRA三分部	Acupuncture / Chinese herbal medicine / dispensing；公开登记与限制条件	只开放注册分部适配服务
PII / PL	保额、除外、追溯与run-off、上门/针灸/手法承保批条	到期或不承保关闭上门；不虚构500万/1000万统一法定额
Provider认可	基金×执业者×地点×模式×有效期×书面依据	无认可不标可返还，仍可真实自费
中药与礼盒	供应主体、ARTG路径或具名患者适用豁免、采购发票/批次	禁批量预配套用具名豁免；商品标签不写疗效保证
诊所与废物	实体地址、感控设施、锐器容器/运输/回收协议	领用/回店/处置三段记录
实际排期	固定小时、复核小时、替补与安全值守	无医生确认不收不可退治疗费

签约资源承诺表

每周12小时到店、8小时替补上门、2小时临床复核是排期提案。写明起止日、取消通知、补偿、医疗争议和基金追偿的事实核查流程；临床责任与平台流程责任分别归责，不用免责条款免除实际过错。

临床合作先判断实际雇佣或承包关系；排期、拒单、定价和设备控制不能仅靠合同名称决定劳动关系。移民与院校合作独立立项，海外学历不直接变成澳洲执照。

B | 四基金询证信与HICAPS处理询问

模板分别发送，不用四基金联名替代逐家答复。HICAPS答复处理能力，基金答复认可与benefit规则；本次未发送任何邮件。

Subject: Written clarification – acupuncture recognition, mobile services and third-party administration

To: [Bupa / Medibank / HCF / nib] Provider Operations

Separate copy for relevant processing questions: HICAPS Compliance Team

Dear Provider Operations Team,

We are preparing a single-clinic Sydney pilot for Australia China Health, a bilingual booking, scheduling and care-coordination platform. The treating practitioner and clinical supplier remain identifiable and responsible for the services actually provided. The platform does not determine benefit eligibility or represent that rebates are guaranteed.

Provider legal name: [Provider Name]

AHPRA registration/division: [Registration Number and Division]

Recognised provider number(s): [Provider Number]

Registered practice address: [Practice Address]

Clinical trading/legal entity and ABN: [Trading Entity / ABN]

Platform entity and ABN: [Platform Entity / ABN]

Contact: [Contact / Email]

Please provide written answers, the effective date, and the governing provider agreement or policy provisions for the following:

1. Mobile acupuncture: Are benefits payable for clinically appropriate acupuncture delivered at a patient's private residence? Please confirm accepted initial/standard item codes, including whether HICAPS 103/203 are applicable. What mobile recognition and location registration are required? Is a distinct provider number required, and what address must appear on the receipt? We will record the actual treatment address and will not describe an in-home service as clinic attendance.
2. Travel and administration: May a separately disclosed travel/administration charge be billed as non-claimable? Please specify whether a clearly itemised single receipt is acceptable when one entity supplies both items, and whether distinct suppliers must issue their own receipts. Confirm how taxes, non-claimable items and linked order references should be presented.
3. Claim processing: Which ancillary modalities can currently be processed using HICAPS mobile hardware or other electronic channels? Where point-of-service processing is unavailable, are paid, itemised receipts accepted for patient self-claims? Please specify evidence requirements and any mobile-service exclusions. We do not assume ECLIPSE is an ancillary acupuncture claiming channel.
4. Platform administration: The platform may charge the clinic a separately invoiced technology/coordination fee, proposed as AUD 15 or AUD 35 excluding GST per completed encounter. It will not inflate a clinical item or appear as a clinical provider. Do your recognition terms restrict third-party administrative access, receipt generation, collection agency arrangements, fee sharing or claims submission? What authorisations and audit records are required?
5. Same-day services: What restrictions apply to acupuncture and remedial massage for the same patient on the same day, delivered by the same practitioner or different practitioners? Please identify the policy basis; we will not split one treatment artificially to seek additional benefits.

6. Corrections and refunds: Please specify credit/adjustment note requirements, notification of any previously claimed benefits, and retention/audit expectations for cancellations, partial refunds and factual corrections. We will preserve original receipts and linked correction history.

We seek your operational requirements within the framework of the Private Health Insurance Act 2007, the Private Health Insurance (Health Insurance Business) Rules 2018 (general treatment provisions), the Private Health Insurance (Complying Product) Rules 2015 and applicable private health insurance statement requirements. We understand that legislation and PHIS information do not by themselves establish practitioner recognition, mobile eligibility or a patient's available benefit.

Please distinguish fund recognition from HICAPS processing capability. No patient or practitioner information beyond the minimum necessary will be supplied without appropriate authority.

Yours faithfully,
[Authorised Signatory]
[Trading Entity]
[Contact / Email]

C | 60人深访逐字脚本与承诺验证

招募与分组

肌骨白领、持续睡眠困扰、照护长者子女各20人。至少一半来自诊所外部；按服务地区与近12月消费分层，避免只有熟人和高消费患者。访谈约45-60分钟；自愿参与、可退出，录音另行同意。研究补贴不得抵扣预订以伪造购买。

01 | 过去行为（15分钟）

“过去12个月，你最后三次去GP、Physio或中医分别什么时候？每次标价、保险返还、真正自付多少？可以自愿查看账单，但不用上传身份资料。”

“谁安排预约？你来回花多久，停车多少？有没有取消或忍着不去的一次？当时为什么？”

“最后一次为父母付医药或接送费，具体支付多少、花多久？父母本人对上门是否同意？你愿意代付不等于患者愿意接受治疗。”

02 | 服务理解（5分钟）

展示中性服务说明，不展示价。让受访者复述：治疗时间、其他时间、谁交付、何时复评、礼盒是否可选。询问“哪项你会去掉？”而不是“服务好不好”。睡眠支持不能替代适当的GP、心理或CBT-I转介。

03 | PSM四问（10分钟）

到店与上门分别问：①便宜到担心质量的价格；②便宜而合理的价格；③贵但仍可能考虑的价格；④贵到完全不考虑的价格。记录原始AUD金额，顺序随机轮换，不先讲130/230。正常顺序应太便宜≤便宜≤贵≤太贵，矛盾时追问理解而不替患者改数字。

每SKU绘制四条累计分布曲线；交点是样本描述，不是60人的统计代表性保证。报告样本构成、缺失、异常值、区间和实际预订率，不能只给一个“最优价”。

04 | 零私保与会员（10分钟）

“现在知道到店130、上门230，私保返还为零。你会选择哪一个、继续原服务、还是都不选？下一次出现什么具体情况你才会购买？”

“如果你已经有可约的诊所，为什么还要多付100上门？能省掉哪些实际成本？如果是年假或远程办公，机会成本是否真的存在？”

“792年卡最多4人，全户300分钟协调，不含针灸中药。你可以举一个过去发生、需要这些协调的任务吗？谁最终付款？你为什么不买？”

“额度用完，15分钟22，你会付还是换别的方式？为什么？若只想咨询医生，这张协调卡可能不适配。”

05 | 防口头支持与真实动作（5分钟）

“你刚才说愿意付230，上一次类似问题却只付80。差别是什么？如果今天无需购买，你是否只是支持项目？”

只有诊所有真实可订排期时展示预约：20可退定金、明确预约/全价/取消条件，允许立即退款及拒绝。没有真实排期时只做无扣款预约模拟，不记录为付费承诺。

记录：看价卡→选择时段→患者本人授权→真实定金→完成全价履约→退款原因。将口头意愿、订金和完成分别计数，不合并为转化。收款和联系前取得相应授权。

数据字典

研究ID、分组、地区、近12月自付、交通分钟/停车、四个PSM价格、零报销选择、家庭会员选择、真实预订步骤、退出理由。使用去标识记录；不把研究消费数据自动写入临床病历或营销名单。

D | PO数据库、RLS与三状态机

状态机	主要状态	关键异常与规则
预约	requested→screened→accepted→confirmed→arrived→completed; aborted/cancelled/expired分支	医生/患者重新确认改期；安全离场独立于患者评分
支付	not_authorised→authorised→capture_pending→paid; part_refunded/refunded/failed/disputed/expired	授权到期重新处理；验签、幂等、回查；退款失败留工单
凭据	报价→真实履约签核→已付收据→贷项/调整→净额版本	不伪造治疗日；旧件不可覆盖；基金已申领需通知调整

服务端写入口规则

预约用事务锁+医生时间段冲突检查；金额、道路距离、资格和同意版本从服务端证据读取。任何客户端成功跳转不确认付款。Webhook验签、event_id唯一、幂等锁、累积退款回查；安全与临床完成不因弱网丢失。

父母本人授权按calendar/reschedule/diary分开；临床记录不由代付身份获得。所有浏览器写权限关闭，由受限后端实现审查后的状态转换。临床笔记AES-GCM密文、密钥外置；管理员读取先提交审计再解密。不得将service-role密钥或解密密钥放进HTML。

下列为新版本核心DDL与读权限骨架，未实现完整预约写事务、积分配额及生产部署。代码与展示原型均不能当作已上线医疗系统。

```
-- ACH P0 architecture baseline. Apply in a clean Supabase project.
-- Read permissions are testable here. Lifecycle writes need reviewed server transactions.
begin;
create schema if not exists private;
revoke all on schema private from public, anon;
grant usage on schema private to authenticated;
create table public.staff_roles(user_id uuid primary key references auth.users, role text not null check(role in ('operations','practitioner','clinical_admin')));
create table public.patients(id uuid primary key default gen_random_uuid(), user_id uuid unique not null references auth.users, full_name text not null, dob date not null, contact jsonb not null, preferred_language text not null check(preferred_language in ('zh','en')), emergency_contact jsonb not null, last_service_at timestamptz, retain_until date, legal_hold boolean not null default false);
create table public.practitioners(id uuid primary key default gen_random_uuid(),user_id uuid unique not null references auth.users,name text not null,ahpra_reg_number text unique not null,ahpra_division text[] not null,pii_expiry date not null,mobile_approved boolean not null
```

```

default false,status text not null check(status in ('review','verified','suspended')));
create table public.provider_recognitions(id uuid primary key default
gen_random_uuid(),practitioner_id uuid not null references public.practitioners,fund text not
null check(fund in ('Bupa','Medibank','HCF','nib')),provider_number text not null,service_mode
text not null check(service_mode in ('clinic','home')),practice_address text not
null,valid_from date not null,valid_until date not null,evidence_ref text not
null,check(valid_until>=valid_from));
create table public.appointments(id uuid primary key default gen_random_uuid(),patient_id uuid
not null references public.patients,practitioner_id uuid not null references
public.practitioners,service_type text not null check(service_type in
('sleep_support','musculoskeletal')),service_mode text not null check(service_mode in
('clinic','home')),slot_start timestampz not null,slot_end timestampz not null,status text
not null check(status in
('requested','screened','accepted','confirmed','arrived','completed','aborted','cancelled','exp
ired')),treatment_address text not null,road_distance_km numeric(8,3),clinical_cents integer
not null,travel_cents integer not null,clinical_gst_cents integer not null,travel_gst_cents
integer not null,idempotency_key uuid not null
unique,check(slot_end>slot_start),check(clinical_cents>=0 and
travel_cents>=0),check(service_mode<>'home' or road_distance_km between 0 and 10));
create index appointments_by_patient on public.appointments(patient_id);
create index appointments_by_practitioner on public.appointments(practitioner_id);
create table public.consents(id uuid primary key default gen_random_uuid(),patient_id uuid not
null references public.patients,consent_type text not null check(consent_type in
('treatment','health_data','care_team','marketing')),version text not null,content_hash text
not null,granted boolean not null,granted_at timestampz not null default now(),ip_address
inet);
create table public.patient_grants(id uuid primary key default gen_random_uuid(),patient_id
uuid not null references public.patients,grantee_user_id uuid not null references
auth.users,scope text not null check(scope in ('calendar','reschedule','diary')),granted_at
timestampz not null default now(),expires_at timestampz not null,revoked_at
timestampz,check(expires_at>granted_at));
create table public.payments(id uuid primary key default gen_random_uuid(),appointment_id uuid
not null references public.appointments,payer_user_id uuid not null references
auth.users,status text not null check(status in
('not_authorised','authorised','capture_pending','paid','part_refunded','refunded','failed','di
sputed','expired')),stripe_payment_intent text unique,amount_cents integer not null
check(amount_cents>=0),refunded_cents integer not null default 0,check(refunded_cents between 0
and amount_cents));
-- Only a generic payer statement is exposed: no service type, symptoms or treatment address.
create table public.payer_statements(id uuid primary key default
gen_random_uuid(),payer_user_id uuid not null references auth.users,order_reference text not
null,total_cents integer not null,status text not null);
create table public.invoices(id uuid primary key default gen_random_uuid(),appointment_id uuid
not null references public.appointments,invoice_number text unique not null,document_kind text
not null check(document_kind in ('medical','travel','credit','adjustment')),original_invoice_id
uuid references public.invoices,provider_abn text not null,provider_number text,provider_name
text not null,ahpra_number text,practice_address text not null,treatment_address text not
null,patient_name text not null,service_date date not null,item_codes jsonb not
null,subtotal_cents integer not null,travel_cents integer not null,gst_cents integer not
null,total_cents integer not null,payment_status text not null,issued_at timestampz not null
default now(),reason text,check(document_kind not in ('credit','adjustment') or
original_invoice_id is not null));
create table public.clinical_notes(id uuid primary key default gen_random_uuid(),appointment_id
uuid not null references public.appointments,ciphertext bytea not null,nonce bytea not
null,auth_tag bytea not null,key_version integer not null);
create table public.audit_log(id bigint generated always as identity primary key,actor_user_id
uuid not null,event_type text not null,object_id uuid not null,reason text not null,created_at
timestampz not null default now());
create table public.webhook_events(event_id text primary key,object_id text not null,event_type
text not null,payload_hash text not null,received_at timestampz not null default
now(),processed_at timestampz);
create table public.membership_usage(id uuid primary key default gen_random_uuid(),family_id

```

```

uuid not null,actor_user_id uuid not null references auth.users,minutes integer not null
check(minutes>0),purpose text not null,occurred_at timestamptz not null default
now(),idempotency_key uuid not null unique);

-- Helper owner must be controlled by DB administration; never callable with an arbitrary
actor.
create function private.owns_patient(p uuid) returns boolean language sql stable security
definer set search_path='' as $$select exists(select 1 from public.patients where id=p and
user_id=auth.uid())$$;
create function private.has_grant(p uuid,s text) returns boolean language sql stable security
definer set search_path='' as $$select exists(select 1 from public.patient_grants where
patient_id=p and grantee_user_id=auth.uid() and scope=s and revoked_at is null and
expires_at>now())$$;
create function private.assigned_patient(p uuid) returns boolean language sql stable security
definer set search_path='' as $$select exists(select 1 from public.appointments a join
public.practitioners r on r.id=a.practitioner_id join public.staff_roles s on
s.user_id=r.user_id where a.patient_id=p and r.user_id=auth.uid() and r.status='verified' and
s.role='practitioner' and a.status in ('accepted','confirmed','arrived','completed'))$$;
create function private.treats_appointment(p uuid) returns boolean language sql stable security
definer set search_path='' as $$select exists(select 1 from public.appointments a join
public.practitioners r on r.id=a.practitioner_id join public.staff_roles s on
s.user_id=r.user_id where a.id=p and r.user_id=auth.uid() and s.role='practitioner' and
r.status='verified' and a.status in ('accepted','confirmed','arrived','completed'))$$;
create function private.owns_appointment(p uuid) returns boolean language sql stable security
definer set search_path='' as $$select exists(select 1 from public.appointments a join
public.patients t on t.id=a.patient_id where a.id=p and t.user_id=auth.uid())$$;
create function private.is_clinical_admin() returns boolean language sql stable security
definer set search_path='' as $$select exists(select 1 from public.staff_roles where
user_id=auth.uid() and role='clinical_admin')$$;

alter table public.staff_roles enable row level security;
alter table public.patients enable row level security;
alter table public.practitioners enable row level security;
alter table public.provider_recognitions enable row level security;
alter table public.appointments enable row level security;
alter table public.consents enable row level security;
alter table public.patient_grants enable row level security;
alter table public.payments enable row level security;
alter table public.payer_statements enable row level security;
alter table public.invoices enable row level security;
alter table public.clinical_notes enable row level security;
alter table public.audit_log enable row level security;
alter table public.webhook_events enable row level security;
alter table public.membership_usage enable row level security;
alter table public.patients force row level security;
alter table public.appointments force row level security;
alter table public.consents force row level security;
alter table public.patient_grants force row level security;
alter table public.payments force row level security;
alter table public.invoices force row level security;
alter table public.clinical_notes force row level security;
alter table public.audit_log force row level security;
alter table public.membership_usage force row level security;

revoke all on
public.staff_roles,public.patients,public.practitioners,public.provider_recognitions,public.app
ointments,public.consents,public.patient_grants,public.payments,public.payer_statements,public.
invoices,public.clinical_notes,public.audit_log,public.webhook_events,public.membership_usage
from anon,authenticated;
grant select on
public.patients,public.consents,public.patient_grants,public.payer_statements,public.invoices
to authenticated;

```

```

grant select(id,patient_id,practitioner_id,slot_start,slot_end,status) on public.appointments
to authenticated;
create policy patient_read on public.patients for select to authenticated
using(user_id=auth.uid() or private.assigned_patient(id));
create policy appointment_calendar on public.appointments for select to authenticated
using(private.owns_patient(patient_id) or private.has_grant(patient_id,'calendar') or
private.treats_appointment(id));
create policy consent_owner on public.consents for select to authenticated
using(private.owns_patient(patient_id));
create policy grant_visible on public.patient_grants for select to authenticated
using(private.owns_patient(patient_id) or grantee_user_id=auth.uid());
create policy payer_own on public.payer_statements for select to authenticated
using(payer_user_id=auth.uid());
create policy invoice_clinical on public.invoices for select to authenticated
using(private.owns_appointment(appointment_id) or private.treats_appointment(appointment_id));

-- No direct SELECT on notes, including admins. This returns ciphertext only.
-- Backend commits audit BEFORE external-key decryption; it validates auth.getUser().
create function private.audited_note_read(n uuid,why text) returns table(ciphertext bytea,nonce
bytea,auth_tag bytea,key_version integer) language plpgsql security definer set search_path=''
as $$
declare a uuid;begin
  if auth.uid() is null or length(trim(why))<10 then raise exception 'Authenticated reason
required';end if;
  select appointment_id into a from public.clinical_notes where id=n;
  if a is null or not(private.owns_appointment(a) or private.treats_appointment(a) or
private.is_clinical_admin()) then raise exception 'Access denied';end if;
  insert into public.audit_log(actor_user_id,event_type,object_id,reason)
values(auth.uid(),'clinical_note_read',n,why);
  return query select c.ciphertext,c.nonce,c.auth_tag,c.key_version from public.clinical_notes c
where c.id=n;
end $$;
revoke all on all functions in schema private from public,anon,authenticated;
grant execute on function
private.owns_patient(uuid),private.has_grant(uuid,text),private.assigned_patient(uuid),private.
treats_appointment(uuid),private.owns_appointment(uuid),private.is_clinical_admin(),private.aud
ited_note_read(uuid,text) to authenticated;
-- Provider expiry, slot overlaps, atomic transitions, refund/credit linking and
-- 300-minute aggregate limits are validated by restricted server transactions.
-- Do not expose private schema through PostgREST. No client UPDATE grants here.
commit;

```

测算口径与证据索引

财务模型输入

输入	情景
初始融资	450000一次到账；分批到账另算
订单端点M12/M24/M36	到店150/450/900；上门50/150/300；节点间线性
会员新购	M7-12每月3户，随后6/10/15/20；年续费70%
服务费	到店15，上门35未税；各直接成本5.8/12.8
会员	每月收入60、成本25；支付费实际在续费当月支付
固定费	M1-3每月10k；M4-12每月6k；Y2每月10k；Y3每月15k
启动/复制支出	M1-3各20k，M13为15k，M25为20k，全部费用化
现金处理	GST准备金分开；不含临床流水、所得税、公共采购收入
减半情景	订单及新会员减半；固定费不下降，维持压力测试

临床个体服务GST-free以ATO条件为准。出行、平台服务、会员税务按应税GST登记的情景；诊所贡献暂不计进项GST抵扣。未包含停车、异常出诊及全部固定费用。三次630测算暂拆治疗175+出行35/次，仍须按真实供应与税务决定。

财务收入按提供服务确认；年卡预收减去递延得到保护后的可用现金。模型无折旧、利息和所得税，所有一次性开支费用化，因此EBITDA为本模型简化运营结果，不能当审计利润表。

依据列表 / 2026-10-04研究更新

- S1 · [ABS 2021 Greater Sydney](#)
- S2 · [ABS Patient Experiences 2024-25](#)
- S3 · [MBS针灸服务说明](#)
- S4 · [HCF Healthcover Extras样本](#)
- S5 · [Bupa FLEXtras](#)
- S6 · [Medibank Comprehensive Extras](#)
- S7 · [AHPRA广告规范](#)

- S8 · [TGA临证调配与成药](#)
- S9 · [SafeWork NSW独处工作](#)
- S10 · [IPC NSW健康记录留存](#)
- S11 · [ATO健康GST](#)
- S12 · [Stripe AU定价](#)
- S13 · [Stripe授权期限](#)
- S14 · [Stripe Webhook](#)
- S15 · [HICAPS项目码](#)
- S16 · [VIC Health Records Act](#)
- S17 · [Support at Home服务清单09/2026](#)
- S18 · [Associated provider规则](#)
- S19 · [NDIS不可支付清单](#)
- S20 · [NSW EPA临床废物](#)
- S21 · [CMBA注册标准](#)
- S22 · [OAIC APP6](#)
- S23 · [PHI Health Insurance Business Rules 2018](#)
- S24 · [PHI Complying Product Rules 2015](#)